creative computing

Mar-Apr 1978
vol 4, no 2

Austria Sch 43.00
Belgium BF 94.00
Canada $2.50
Denmark Kr 15.00
Finland Mk 9.00
France F 12.00
Germany DM 6.00
Great Britain £ 1.30
Greece Dr 87.00
Holland Df 17.00
Italy L 2000
Norway Kr 13.00
Portugal Esc 73.00
Spain Ptas 160.00
Sweden Kr 11.00
Switzerland SF 6.00
USA $2.00

the #1 magazine of computer applications and software

$2.00

Business Computing: Inventory Control

Structured Software for Personal Computing

ABCs of Micro-computers

Evaluation of Two BASICs, Micro-APL

Computer Music

Computer Games:
- **Oil Company**
- **Racetrack**

The Great DATAMAZING Parody

CAI: Interaction Between Student and Computer

"Programming is Learned by Practice, Not by Listening"

6800/2 IS HERE

The 6800/2 uses our new A2 processor board with socket space for 8K bytes of ROM/PROM. This makes it possible to use the 6800 in applications where ROM programs are useful without purchasing an expensive PROM accessory board. The A2 board has a DIP switch selector that allows you to replace any 8K block of memory above the RAM memory that extends to 32K with memory external to the processor board itself. This lets you develop special programs that will later be put in PROM in a normal RAM memory card where it can be modified and debugged. The A2 board has a crystal controlled baud rate oscillator and a separate clock driver oscillator whose frequency may be changed with a programming resistor. The A2 processor board gives you the maximum possible flexibility in setting up a computer system.

SWTBUG® Monitor—

The 6800/2 is supplied with our new SWTBUG® monitor. This new monitor is software compatible with the earlier Mikbug® monitor used in the 6800. All major subroutine entry points are identical. SWTBUG® features a resident MF-68 Minifloppy disk boot, single level breakpoints, vectored software interrupt, generation of punch end of tape formatting and automatic interface configuring for either the MP-C control interface or MP-S serial interface.

ACIA Type Interface—

The 6800/2 uses our MP-S serial interface. This RS-232 and 20 Ma. TTY compatible interface may be configured to operate serially at the following baud rates: 110, 150, 300, 600, 1200, 2400, 4800 and 9600. Complete interrupt control is available through the user's software.

4K Static MEMORY—

The 6800/2 comes wth 4K of static RAM memory on our MP-8M board. The memory may be expanded to 8K by the addition of eight more memory chips. No additional parts are needed. Full buffering of all data, address and control lines is a standard feature. Memory expansion to 32K of continuous RAM memory and up to a 48K mixture of ROM/RAM is possible with this system.

ACCESSORY BOARDS—

Do you have a special job? Our accessory boards make it possible to use the 6800/2 for almost any type of computer application. We have our MP-T interrupt timer with software interrupt selectable output. Our MP-N calculator interface that allows you to do arithmetic functions in hardware. Our MP-R EPROM programmer that programs and verifies EPROMs right in the machine—and more coming.

6800/2 Kit. . . . . **$439.00 ppd Cont. U.S.**
6800/2 Assembled. . . . . **$495.00 ppd Cont. U.S.**

SOUTHWEST TECHNICAL PRODUCTS CORPORATION
219 W. RHAPSODY
SAN ANTONIO, TEXAS 78216

PHOENIX, AZ
BYTE SHOP OF ARIZONA

LITTLE ROCK, AR
COMPUTER PRODUCTS UNLTD.

CARSON, CA
SUNSHINE COMPUTER CO.

CITRUS HEIGHTS, CA
BYTE SHOP OF SACRAMENTO

LOOMIS, CA
KINGMONT ENTERPRISES INC.

SAN MATEO, CA
COMPUTER TERMINAL

SUNNYVALE, CA
RECREATIONAL COMPUTERS

VAN NUYS, CA
COMPUTER COMPONENTS

ATLANTA, GA
ATLANTA COMPUTER MART

HAIKU, MAUI, HI
CAPACITY, INC.

HONOLULU, HI
COMPACT COMPUTERS

KAILUA, OAHU, HI
MICROCOMPUTER ENTERPRISES

SCHAUMBURG, IL
DATA DOMAIN

BLOOMINGTON, IN
DATA DOMAIN

FORT WAYNE, IN
DATA DOMAIN

PIKESVILLE, MD
MODULAR SYSTEMS, INC.

ROCKVILLE, MD
COMPUTER WORKSHOP

BURLINGTON, MA
THE COMPUTER STORE

EDINA, MN
COMPUTER DEPOT

ISELIN, NJ
COMPUTER MART OF NEW JERSEY

EGGERTSVILLE, NY
CORSON COMPUTER CORP.

HOLLIS, NY
SYNCHRO SOUND ENTERPRISES

CINCINNATI, OH
DATA DOMAIN

DAYTON, OH
DATA DOMAIN

COLUMBIA, SC
BYTE SHOP OF COLUMBIA

NASHVILLE, TN
GURYA CORP.

CORPUS CHRISTI, TX
MICRO SYSTEMS SERVICES, INC.

DALLAS, TX
COMPUSHOP

HOUSTON, TX
ELECTROTEX

HOUSTON, TX
MICROCOMPUTER ORIGIN STORE

MADISON, WI
THE MADISON COMPUTER STORE

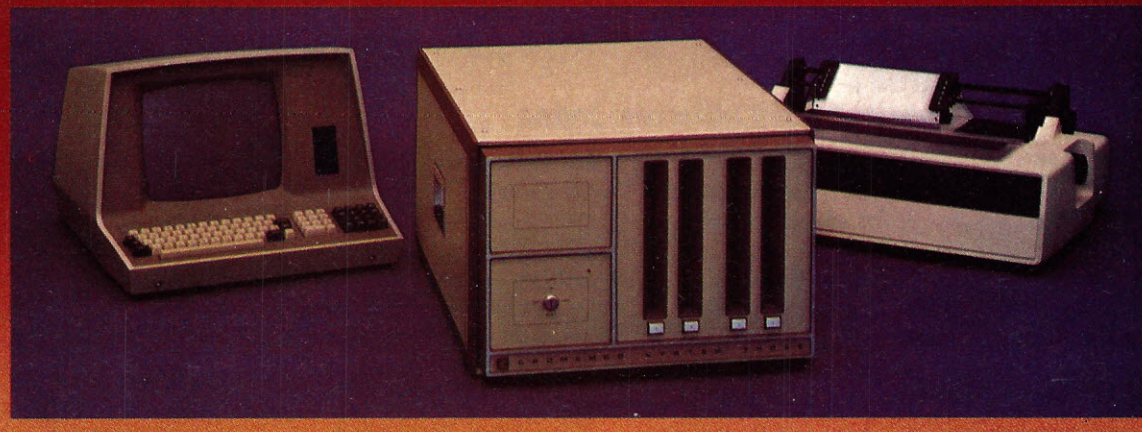
HUNTINGDON, CAMBS, ENGLAND
COMART LTD

WEDEL (HOLSTEIN), GERMANY
DIGITRONIC COMPUTER SYSTEM GmbH

STOCKHOLM, SWEDEN
DATORISERING KONSULT AB

ZURICH, SWITZERLAND
COMICRO AG

Now we can announce it– the multi-disk drive System Three Computer



A fast Z80 microcomputer with up to 512 kilobytes of RAM, 4 disk drives and 1 megabyte of disk storage—with CRT terminal and fast printer. Even an optional PROM programmer. Strong software support, too, like FORTRAN IV, Extended BASIC, and Macro Assembler.

PROFESSIONAL GRADE— FOR PROFESSIONALS

Chances are you've already heard that there is a Cromemco System Three Computer. We've proudly previewed it at WESCON on the West Coast and NYPC on the East Coast.

It's a complete system—processor, CRT terminal, line printer.

First it's fast—1 microsecond nominal execution time and 250 nanosecond cycle time.

Its equally fast RAM memory is large and enormously expandable—32 kilobytes expandable to 512 kilobytes. No danger of obsolescence from inadequate RAM capacity.

THE ONLY MICROCOMPUTER OFFERING 4 DISK DRIVES

Further, the System Three comes with two disk drives to give you 512 kilobytes of disk storage. Soft-sectored IBM format. Optionally, you can have four drives with 1 megabyte of storage.

There's disk protection, too, since in the LOCK position disks can't be ejected while they are running.

21-SLOT MOTHERBOARD

This new CS-3 is a computer that won't be outdated soon. It has a 21-card-slot slide-out motherboard and an S-100 bus so that you can plug in all sorts of support circuitry. The heavy-duty 30-amp power supply can easily handle all this.

BROAD S-100 SUPPORT

The S-100 is the bus that Cromemco so strongly supports with over a dozen plug-in circuits ranging from analog I/O to high-speed RAM memory with our bank-select feature.

TRULY POWERFUL SOFTWARE

You have to have software. And Cromemco is far in front there, too. Our FORTRAN IV, for example, is equal to the FORTRAN compilers on large mainframes. Further, it (and our other software) is low-priced.

Our 16K Z80 BASIC is one of the fastest and most capable. Full 14-digit precision.

There's also our Z80 Macro Assembler and Linking Loader. Uses Z80 mnemonics. Allows referencing FORTRAN common blocks.

SEE AT YOUR DEALER

You have to see the CS-3 to fully appreciate it and its low prices starting at $5990 in the rack mount version.

Better contact your dealer now.

Cromemco
incorporated
Specialists in computers and peripherals
2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

CIRCLE 102 ON READER SERVICE CARD

"The 4051 enables an ideal learning technique: one-to-one dialogue with graphic examples."

Alfred Bork
University of California, Irvine

Imagine a teaching assistant who continually involves students with intriguing graphic demonstrations. Who stays close enough to each student to critique answers and review material immediately after testing. Who tutors according to individual interests and learning rates.

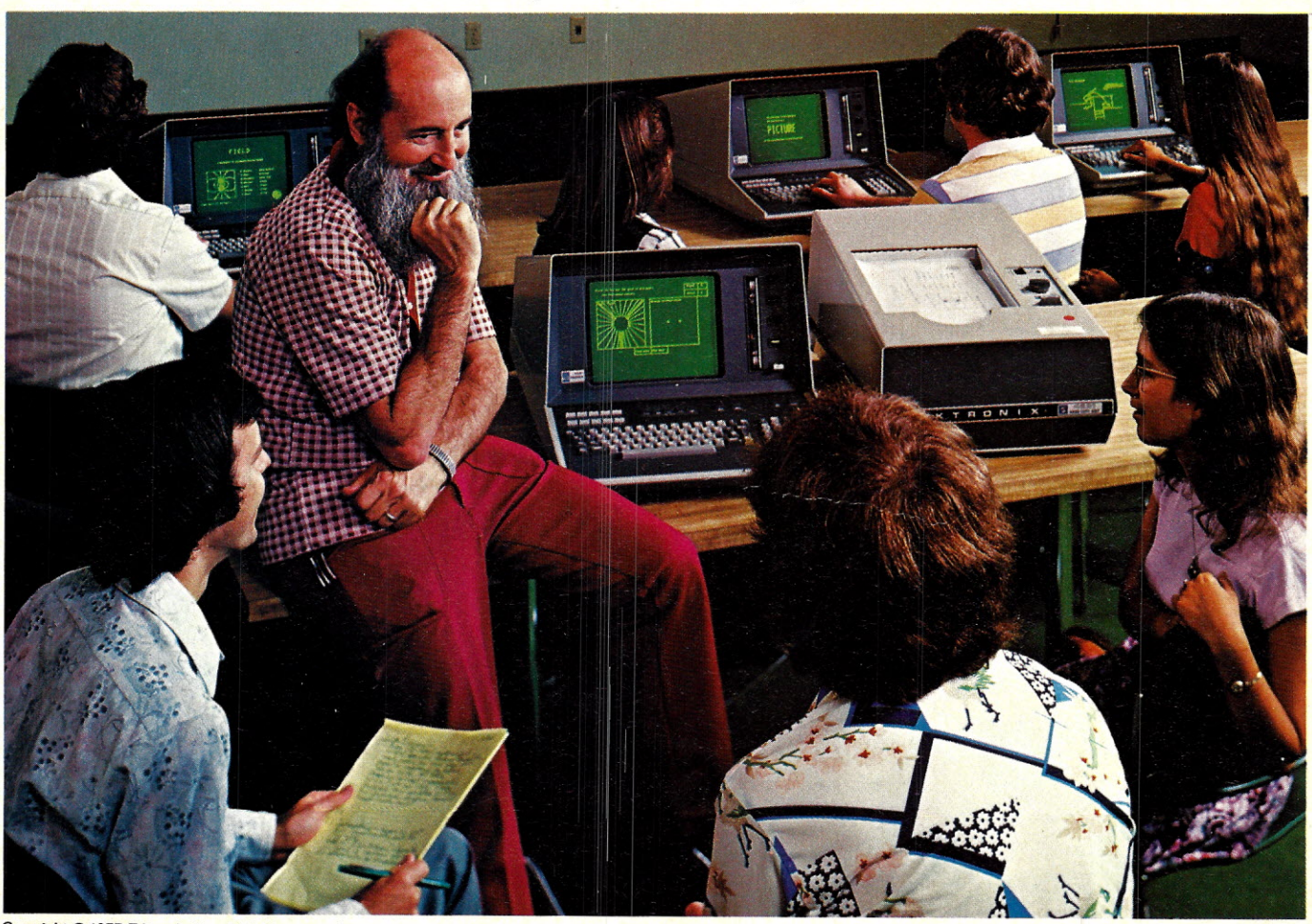
Dr. Alfred Bork brings that kind of assistance into his physics classes.

It's Tektronix Computer Graphics. For example, the 4051, pictured here, works as a low-cost, off-line or on-line learning device to help students study energy, gravity, momentum, and dozens of other subjects via fascinating simulations and graphic metaphors. It's a big assist to self-paced methods. To testing. Even to course management.

Whatever your subject matter, whoever your students, the 4051 makes learning a very memorable experience. For a video tape on Computer Graphics in Learning, application or product literature, please write: Institutional Market Manager

Tektronix, Inc.
Information Display Group
P.O. Box 500
Beaverton, Oregon 97077
Tektronix Datatek NV
P.O. Box 159
Badhoevedorp
The Netherlands

Tektronix®
COMMITTED TO EXCELLENCE



CIRCLE 116 ON READER SERVICE CARD

in this issue . . .

articles

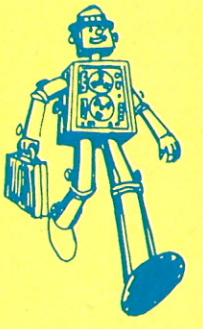
BUSINESS COMPUTING: INVENTORY CONTROL

review & resources

fiction & foolishness

things to do - games

departments



The Cover

An encounter of the fourth kind — businessmen encounter a computer and (1) expect it will solve all their problems, (2) regard it as a mystical and magical machine and (3) expect it to eat punched cards. Today it is none of these things and some of the Real Truth is revealed in the pages of this issue of *Creative*.

The title of the painting is "Mendelov Conspiracy" by Paul Stinson. Publishers often have similar tastes; as a result the painting will also appear shortly on the cover of the book *Encounter Three* by Martin Canon, published by Pinnacle Press.

Illustration copyright© 1978 by Paul Stinson.

Mar-Apr 1978 Volume 4, Number 2 Consecutive Issue No. 20

Creative Computing magazine is published bi-monthly by Creative Computing, P.O Box 789-M, Morristown, NJ 07960. (Editorial office: 51 Dumont Place, Morristown, NJ 07960. Phone: (201) 540-0445.)

Domestic Subscriptions: Institutional 1-year $15, 3-year $40; individual 1-year $8, 2-year $15, 3-year $21. Subscription orders, change of address, P.O. Form 3579 to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. Call 800-631-8112 toll-free to order a subscription (to be charged only to a bank card).

Foreign Subscriptions:

Great Britain. Institutional 1-year £9.40, 2-year £11.90, 3-year £25.00; Individual 1-year £6.25, 3-year £16.90. Orders to Creative Computing, 60 Portchester road, Southampton SOZ7JD, England.

Europe, 1-year subscription. Austria Sch 180, Belgium BF 4000, Denmark Kr65, Finland Mr40, France Fr50, Germany DM25, Greece Dr360, Holland Dfl27, Italy L8500, Norway Kr55, Portugal Esc320, Spain Ptas700. Sweden Kr45, Switzerland SF 25. Orders to Pan Atlantic Computer Systems GmbH, Frankfurter Strasse 78, D61 Darmstadt, German Fed Rep.

Other Countries 1-year $12, 2-year $23, 3-year $33 (surface postage, U.S. dollars). Orders to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

Second class postage paid at Morristown, New Jersey and at additional mailing offices.

Publisher
David H. Ahl

Editor-in-Chief
Stephen B. Gray

Managing Editor
Burchenal Green

Associate Editor
Steve North

Higher Education Editors
Merl Miller
A. Kent Morton

Technology Editor
Alan Salisbury

Careers Editor
Eleanor Corr

Humor Editor
Jack Ludwig

Art Director
Elyse J. Fox

Marketing Manager
Linda Harrison

Bookkeeper
Jeanne Tick

Retail Marketing
Linda Eckerstrom

Customer Service
Ethel Fisher

Subscriptions
Nancy Hammond
Carol Cassata

Book Service
Barbara Shupe
Katherine McKenzie

Order Processing
Thomas Bass

Advertising Sales

Western States, Texas
Jules E. Thompson
Hearst Building, Suite IIII
5 Third Street
San Francisco, CA 94103
(415) 362-8547

Southern California
Bert Charlton
2560 Via Tejon
Palos Verdes Estates, CA 90274
(213) 378-8361

Mid-Atlantic, Northeast
Charles Lynch
36 Sohier Street
Cohasset, MA 02025
(617) 383-6136

Elsewhere
Burchenal Green
(201) 540-0445

MEMBER
COSMEP
COMMITTEE OF SMALL MAGAZINE EDITORS AND PUBLISHERS

...notices...

Personal Computing Festival

The 1978 National Computer Conference, to be held in the Anaheim Convention Center, June 5-8, will include a full-scale, three-day **(June 6-8)** Personal Computing Festival in the nearby Disneyland Hotel.

The Festival will include paper, panel and tutorial sessions; a contest for microprocessor systems and applications; and a commercial exhibit of personal computing products and services.

Marie Stewart, National Computer Conference, c/o AFIPS, 210 Summit Ave., Montvale, NJ 07645. (201) 391-9810.

AEDS Computer Programming Contest

The Association for Educational Data Systems (AEDS) has announced its Fifteenth Annual Computer Programming Contest for students in grades 7-12. Deadline for entries in the contest is **March 1, 1978.** Entries may be submitted in the following categories: business, biological and physical sciences, computer art, computer science, games, humanities, and mathematics.

The Grand Prize winner will receive a $100 U.S. Savings Bond plus a minimum $300 travel grant for travel to the 1978 AEDS Convention in Atlanta Georgia on May 15-19, 1978. Winning student's sponsor will also receive an all-expense-paid trip to the convention. Category winners will receive a $50 U.S. Savings Bond.

Ben Jones, AEDS Programming Contest, OTIS, 1200 Highway 99N, Eugene, OR 97405.

Personal Computing Fair '78

If you're looking forward to nibbling salt-water taffy again at Atlantic City while attending this year's Personal Computing Fair, you'll have to bring it with you, because the location has been changed to Philadelphia. That should make a lot of people happy.

The main dates are the same, **August 25-27, 1978,** at the Civic Center. The 24th has been added as a trade-show date, open to exhibitors and their guests, but not to the public. Headquarters hotel will be the Sheraton.

Current plans include a computer-music concert, and a computer-art show.

Creative Computing will be at the Fair, so come on over and say hello.

John H. Dilks, Fair Director, Personal Computing 78, Rt. 1, Box 242, Mays Landing, NJ 08330.

History of Programming Languages

A History of Programming Languages Conference, sponsored by ACM Special Interest Group on Programming Languages (SIGPLAN) will take place at the Hyatt House Hotel at the Los Angeles International Airport on **June 1-3, 1978,** which is just prior to the 1978 AFIPS National Computer Conference in Anaheim, California.

The purpose of the conference is to create a permanent historical record of the significant events that created the need for the development of the individual languages, of the environment in which decisions were made, and of the rationale behind the decisions which led to the particular language style. In selecting those languages to be discussed, it was decided to limit the choice to those that were deemed most significant *and* which were developed and in use by the end of 1967.

Publicity Chairman: Billy G. Claybrook, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061. (703) 951-5420.

Computing in the Undergraduate Curriculum

The Ninth Conference on Computing in the Undergraduate Curriculum (CCUC/9), will take place **June 12-14** at the University of Denver, Denver, Colorado. All disciplines including agriculture, biological sciences, business, chemistry, computer science, earth sciences, fine arts, history, home economics, humanities, environmental sciences, geography, music, physics, social sciences, psychology, and statistics will be represented.

Conference chairman: Prof. William S. Dorn, Department of Mathematics, University of Denver, Denver, CO 80208. (303) 753-3529.

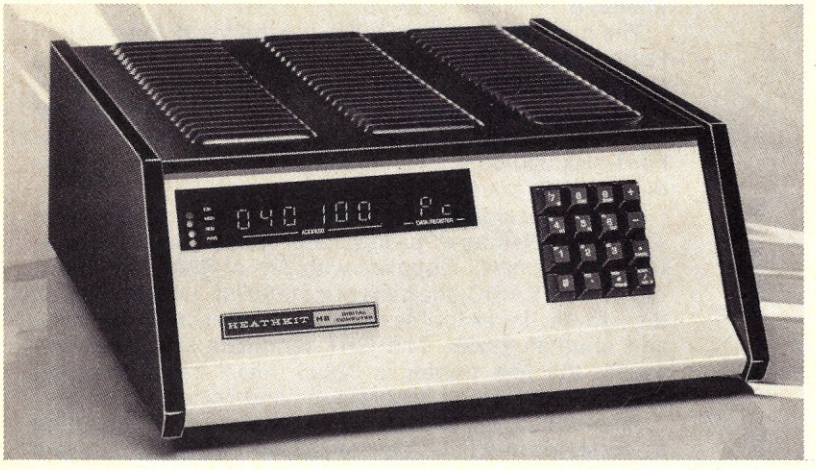
TCF-78

The third annual Trenton Computer Festival will take place on **April 22-23,** at Trenton State College, Trenton, New Jersey.

The program will include 30 speakers/forums, and there will again be a large outdoor flea market. TCF-76 was the first computer festival in this country, and drew 3,000 people. Last year TCF attracted 4,500 people; this year 6,000 are expected.

Co-chairmen: Sol Libes, Dr. Allen Katz, Trenton State College, Trenton, NJ 08625.

This 8-bit machine, by itself, is as versatile as a lot of systems that include peripherals



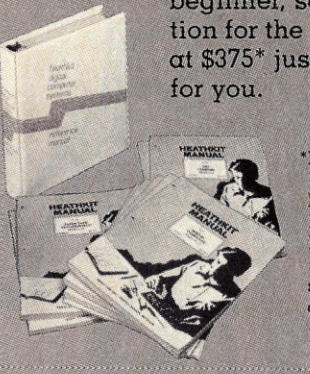
Skeptical? For starters, because of its unique design, the H8 is the only machine in its price class that offers full system integration, yet, with just 4K of memory and using only its "intelligent" front panel for I/O, may be operated completely without peripherals!

Memory Display

Register Display

I/O Port Display

In addition, by using the features of its built-in PAM-8 ROM panel control program, the H8 actually allows you to dig in and examine machine level circuitry. Responding to simple instructions, the "intelligent" panel displays memory and register contents and lets you inspect and alter them even during operation. And for greater understanding, the front panel permits you to execute programs a single instruction at a time. The H8's memory is fully expandable, its 8080A CPU extremely versatile, and with the addition of high speed serial and parallel interfacing you gain the added flexibility of I/O operation with tape, CRT consoles, paper tape reader/punches, and soon floppy disk systems! The H8 offers superior documentation including complete step-by-step assembly and operation manuals, and comes complete with BASIC, assembler, editor, and debug software that others charge over $60 for! H8, simplicity for the beginner, sophistication for the expert and at $375* just right for you.

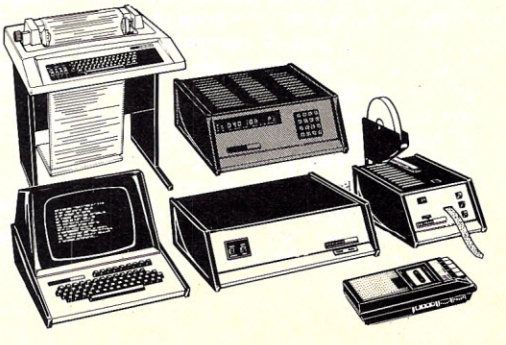


*Prices are mail order net FOB, Benton Harbor, Michigan. Prices and specifications subject to change without notice.

HEATHKIT COMPUTERS

System Engineered for Personal Computing

H8



CIRCLE 105 ON READER SERVICE CARD

Computers, peripherals and nearly 400 exciting, easy-to-build electronic kits, all in your

FREE Heathkit Catalog

HEATH Schlumberger

Heath Company, Dept. 355-390
Benton Harbor, MI 49022

Please send me my FREE Heathkit Catalog.
I am not on your mailing list.

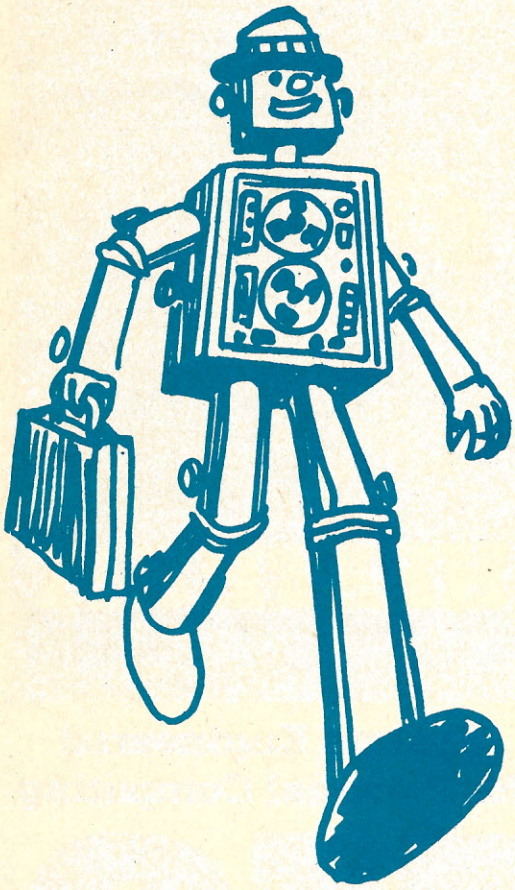
Name_____

Address_____

City_____ State_____

CP-143 Zip_____

al... editorial... editor



Business Computers

It's no secret that for some time more hobby computers have been sold for business use than for hobby use. Go into almost any computer store in the country, and you'll find a business system ready to be demonstrated, or in planning.

The manufacturers are following the trend. Even before Pertec bought MITS, plans were underway to change the latter's marketing orientation from the hobby market to small business systems. Pertec has now introduced the MITS 300 integrated business system, available as the hard-disk 300/55 priced from $15,950 and as the floppy-disk 300/25, priced from $11,450.

MITS is not alone. Cromemco's latest model, the System Three, is billed as a "professional-grade microcomputer for professionals." Supplied with 32K of RAM, FORTRAN IV and 16K BASIC, the base price is $5990, with two floppy disk drives; drives for the other two slots in the main frame add another $2395. The CRT terminals are $1595 and $1995; the line printer is $2995. Imsai has a business system that's priced accordingly, and so on and so on.

All of which is to explain why, in this issue, we're starting a new series of articles devoted to examining the software offered by various types of suppliers for business use. We're starting out with a look at inventory control, giving you first an overview by an independent consultant, and then descriptions of the software offered by several types of suppliers: hardware manufacturers, computer stores, and software houses.

In each issue, from now on, we'll look at a different area of business programs: accounts receivable, accounts payable, order entry, sales analysis, payroll, general ledger, profit-and-loss and balance sheet, medical billing, word processing, etc. In some of these areas, there's not much available at the moment, so we'll hold off on those until enough different programs are on the market so that you'll have a variety to choose from.

Obviously, we can't cover all the suppliers of each type of program. But, since this will be a continuing series, we can touch upon inventory control again, a year or so from now, and take a look at what's new in the market, and do the same with the other types of programs. We welcome any and all input from suppliers of business programs as to present products and future plans.

—Stephen B. Gray

NOW: A Reader Service Card

To make it easier for you to get information on the products offered by our advertisers, we've started including a Reader Service Card in this issue.

No longer do you have to write a letter to each advertiser. Just circle a number on the Reader Service Card, add your name and address, and we'll do the rest.

$24.95 PROBE?

You bet! Meet CSC's Multi-family Logic Probe 2.

Wherever you need fast, safe, accurate digital testing—you need CSC's new LP-2. It's a compact, enormously versatile circuit-powered unit that's become indispensable. As a level detector. Pulse detector. And pulse stretcher.

Easier to use. Set LP-2's switch to the proper logic family, connect two clip-leads to the circuit's supply, touch the probe to the node under test—and you get an instant picture of circuit conditions. Separate LED's indicate logic "1", logic "0", and all pulse transitions. And a 300K-plus input impedance insures minimum circuit loading.

At just $24.95*, you don't have to think twice about owning the LP-2. Especially when you see how it simplifies testing, debugging and servicing all types of digital circuits.

Order today. Call 203-624-3103 (East Coast) or 415-421-8872 (West Coast): 9 a.m.-5 p.m. local time. Major credit cards accepted. Or see your CSC dealer. Prices slightly higher outside USA.

Specifications

Input impedance better than 300KΩ

Thresholds (switch selectable)	**DTL/TTL**	**HTL/CMOS**
logic 1 thresholds (HI-LED)	2.25V ± .10V	70% Vcc ± 10%
logic 0 thresholds (LO-LED)	0.80V ± .05V	30% Vcc ± 10%

Min. detectable pulse width 300nsec.

Pulse detector (PULSE LED) 1/10-sec. pulse stretcher makes high-speed pulse train or single events (+ or − transitions) visible

Input protection overload, ± 25V continuous; 117 VAC for less than 10 sec.; reverse-polarity, 50V

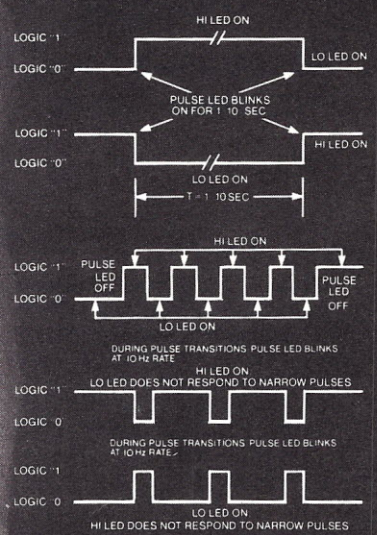
Power requirements 5-15 volts Vcc; 30mA max.

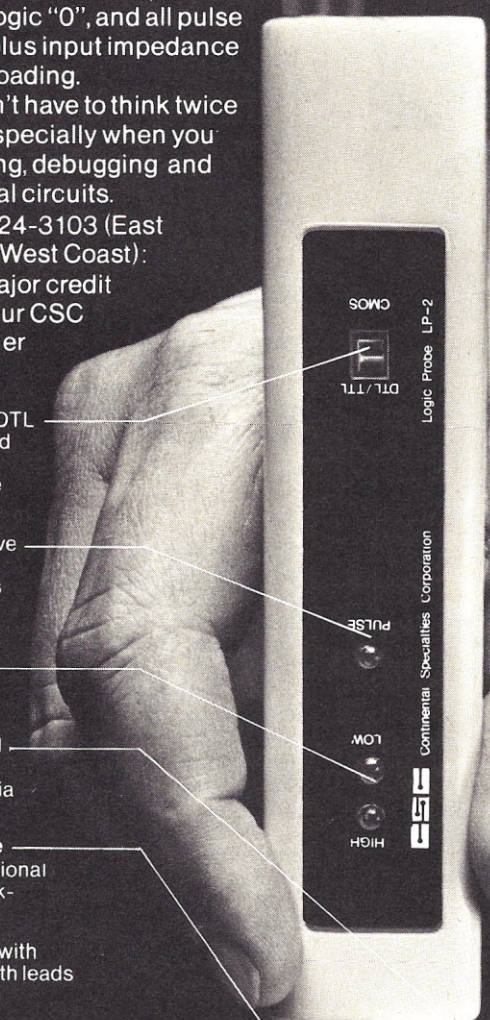
Operating temperature 0-50°C

Physical size (l x w x d) 5.8 x 1.0 x 0.7" (147 x 25.4 x 17.8mm)

Weight 3oz. (.085Kg)

Power leads detachable 24" (610 mm) with color-coded insulated clips; others available





Logic Family Switch—TTL/DTL or CMOS matches Logic "1" and "0" levels for greater versatility. CMOS position also compatible with HTL, HiNIL and MOS logic.

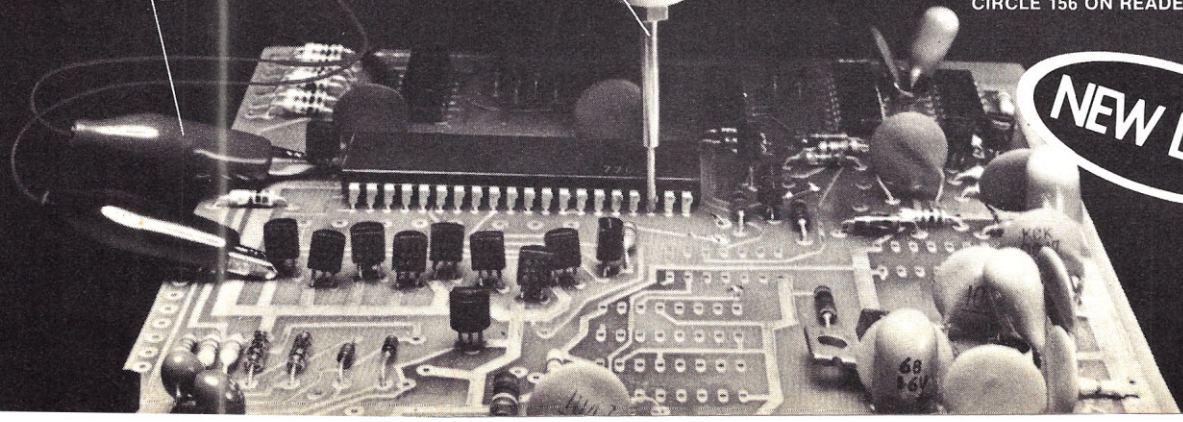
PULSE LED—Indicates positive and negative pulse *and* level transitions. Stretches pulses as narrow as 300 nanoseconds to full 1/10 sec. (10Hz pulse rate).

HI/LO LED's—Display level (HI-logic "1", LO-logic "0") of signal activity.

Interchangeable ground lead connection—Provides ground-side input connection via optional cables.

Interchangeable probe tips—Straight tip supplied; optional alligator clip and insulated quick-connecting clip available.

Plug-in leads—24" supplied, with alligator clips. Virtually any length leads may be connected.



NEW LP-2!

CONTINENTAL SPECIALTIES CORPORATION

CSC

70 Fulton Terrace, Box 1942, New Haven, CT 06509
203-624-3103 TWX 710-465-1227
WEST COAST: 351 California St., San Francisco, CA 94104
415-421-8872 TWX 910-372-7992
GREAT BRITAIN: CSC UK LTD.
Spur Road, North Feltham Trading Estate,
Feltham, Middlesex, England
01-890-8782 Int'l Telex: 851-881-3669

*Manufacturer's Recommended Resale

CIRCLE 156 ON READER SERVICE CARD

put...input/output...in

A Nyce Komplament

Dere Editor:

Thankyou for your publicashun (Jul-Aug issue, p. 141) of your new computor game UFO. It is nice to recieve such an interestin game, and I will make sure my computor is equipted with it soon. The different weapontry is useful for making it a high presure game, especially the lazar. My software library in now aproching completion: all that is needed is an on-line diktionary for spelling and grammer.

Brad Benton
4203 Ramsey
Austin, TX 78756

Help Received

Dear Editor:

This is in reference to my published letter on your July-August 1977 issue asking for help on our PDP-8 system. I have received many helpful replies from your readers, including the letter from John Blake that was published in the Nov.-Dec. issue.

Since our first letter, we have moved our CPU from the third floor to the first floor on the other end of the building and have strung a 24-conductor cable, Beldon #9877, 600 feet to the third floor to service our four ASR33's. We simply cut the Teletype cable to the CPU, added Cinch-Jones connectors to all cable ends, paying attention to color codes, and terminated the 24-conductor cable in a distribution box on the third floor. From the third floor we have six-conductor cables going to four different locations in addition to the four in the room with the distribution box. We have had no problems moving Teletypes (or a DECwriter II) between any of these locations.

Our next project, hopefully, will be to add a non-DEC CRT, and possibly a modern-acoustic coupler arrangement that we can switch from an outside line to our 2-digit phone system. I hope to utilize some of our new contacts for this also!

Dick Brown
Chairman, Science
Minnechaug Regional High School
621 Main St.
Wilbraham, MA 01095

ELIZA

Dear Editor:

Last year [Jul-Aug 1977] you ran an article by Steve North named ELIZA. In it he made a comment about running it on a Southwest 6800, and I took it as both an insult and a challenge.

With some minor restrictions and coding changes, I now have Eliza up and running. Average response time is 15 seconds.

For Southwest owners: the response data must be limited to 32 characters. A little imagination will permit you to so edit the lines. The input string must also be no greater than 32 characters. If you have the 1024 terminal it's simple: one line on the screen.

For everybody: instead of comparing every keyword against the input string, a single character offset at a time, do one search for spaces in the string. Save the offset value + 1 in an array. Then search the string for keywords using the array values for the offset. This will cut down on comparison time. (Assuming a 32-char string, a five-letter word length, original method = 27 compares / keyword * 35 keywords = 945 compares. New method = 32 compares for space + (6 compares / keyword * 35 keywords) = 242 compares.)

Since no rational human being will input a sentence with NOKEYFOUND in it, don't search the string for it. You're wasting time.

If there is no asterisk at the end of the response line, there is no need to conjugate C$. Put the test immediately after the search for keywords. If you fail the test, then conjugate C$.

The program is great for our sceptical friends. I've had some of my wife's friends on it for hours, thus relieving myself of the need to be civil and pseudo-friendly. Try it.

William Stock
1125 Lois Drive
Cincinnati, OH 45237

Steve North replies: Thank you for your comments and suggestions on ELIZA. I'm glad you were able to get it running in SWTPC 8K BASIC. At the time it was written, SWTPC 8K BASIC did not permit character strings longer than 16 characters, but they have since released version 2.0 (which you used) that permits strings of up to 32 characters.

The dummy string NOKEYFOUND is mainly for documentation purposes. If no key word is found, control passes to line 370, which sets the keyword number (variable K) to 36 (which corresponds to NOKEYFOUND) and branches off to line 570, which looks for a reply. No conjugation takes place, as far as I can tell.

In rewriting this program it was not my intention to squeeze out every last byte or nanosecond (though sometimes that's fun). Rather, it seemed more important to come up with something fairly straightforward and understandable. If you feel very adventurous, you could always write a LSIP for the 6800 and ask Weizenbaum for a copy of the original...

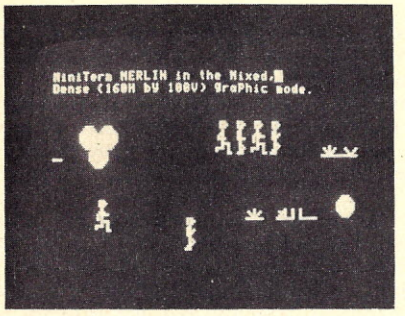
Origins of Heapsort

Dear Editor:

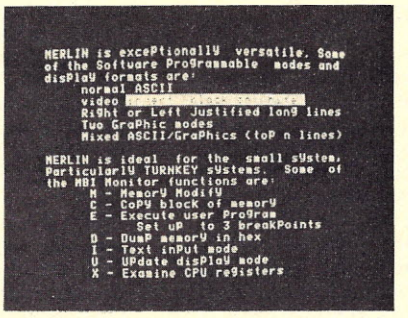
In reference to your article on sorting techniques that appeared a while back, your readers may be interested in the source of that amazing little "Heapsort" Algorithm. I traced it back to the "Communications of the ACM" Vol. 7 #6, June 1964. It was formally presented as Algorithm #232 with references to Algorithm numbers 113, 143, and 144. "Heapsort" has been implemented in various production environments as an in-line disk replacement sort with fantastic results—it really does fly!

R. Vasaly
100 Clarendon Pl., Apt. 9A
Hackensack, NJ 07601

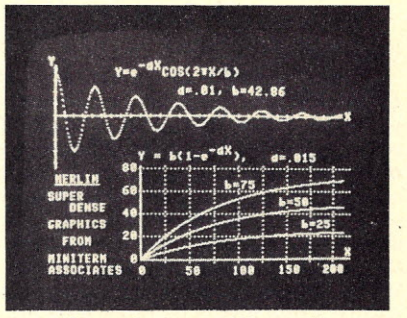
The many faces of MERLIN



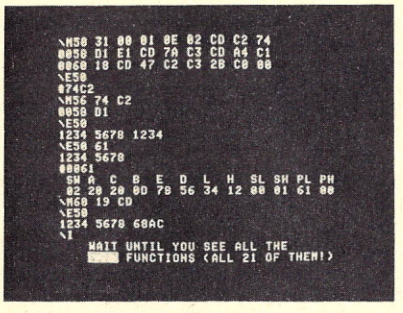
Dense Mode: 160H × 100V
Running Man Patterns



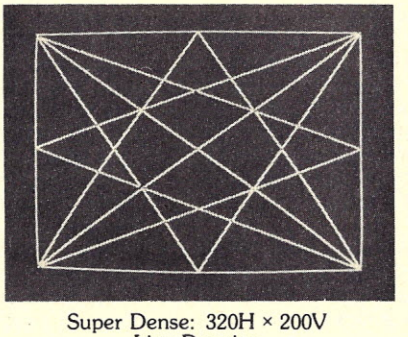
Propaganda



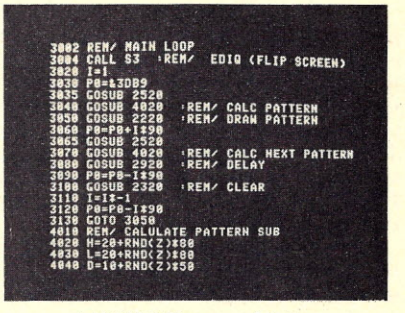
Super Dense: 320H × 200V
Equation Plotting



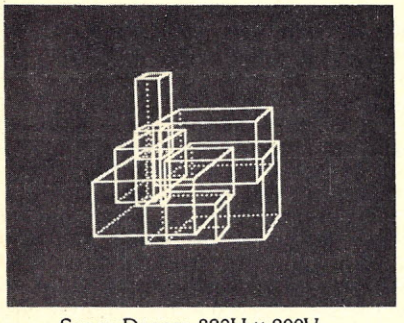
Monitor Debug Usage



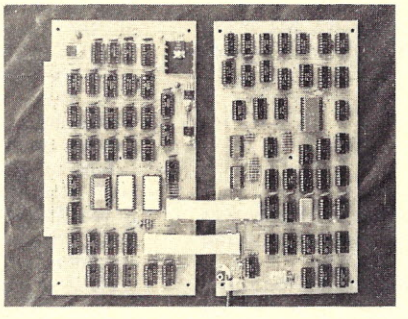
Super Dense: 320H × 200V
Line Drawing

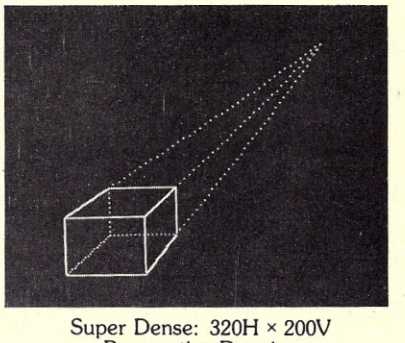


BASIC Program Listing
Output Shown Below



Super Dense: 320H × 200V
3-D Boxes





Super Dense: 320H × 200V
Perspective Drawing

MERLIN (and your S-100 Computer...)
... the graphics development package you've been waiting for.

Do these photos suggest an application in your field? Whether you're into architecture, astrology, music or whatever, you can apply graphics to make your presentations more effective and your work more efficient.

Real Time Plotting
- Heart Rate
- Navigation
- Spectrum Analysis

Complex Equation Plotting
- Stock Market Trends
- Teaching: Mathematics
- Circuit Responses

Fine Line Drawings
- Architecture
- 3-D Projections
- Circuit Layouts

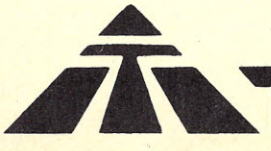
Pattern Movement
- Animation
- Games

MERLIN is also a multipurpose system monitor board with ROM monitor and editor software, parallel keyboard port, and audio cassetts storage, besides being your text and graphics video output device.

MERLIN is supported by expert technical assistance, a constantly expanding series of application notes, and a newly formed users group.

Write or call today for your free copy of our new catalog, an index to available application notes, and a list of **MiniTerm** Dealers.

Assembled and tested **MERLINs** start at $349, $269 for kits. Super Dense add-on, firmware, and cassette interface are extra.



MiniTerm Associates, inc.

Dundee Park, Andover, MA 01810 (617) 470-0525

CIRCLE 106 ON READER SERVICE CARD

put...input/output...in

A Reader Reforms

Dear Editor:

I have decided to renew my subscription to *Creative Computing* and try to salvage something of my life. About two years ago I first saw an issue of *CC*. That issue seemed to be equally devoted to mathematics and computers. While I am quite interested in computers, the mathematical aspects of the magazine were what actually attracted me to subscribe for one year. When the mathematical aspects of the magazine diminished, I decided that I would not renew. I did not realize the profound impact that this decision would have on my life. When the time came for the issues to stop coming, they did not. At first I told myself that they were just issues to bridge the gap in case they had not yet processed a renewal ("they" of course referred to you).

But after a few months I began to realize that I was actually receiving issues I had in no way paid for. I had inadvertently entered a life of crime. These few issues that I was unintentionally stealing from you began to prey upon my mind. At first it started with sleeplessness and rings under my eyes. My job as a systems analyst for the Burroughs Corporation began to suffer. Then the next phase started. I started wearing cheap suits, visiting dives, and talking like James Cagney (or sometimes like Edward G. Robinson). I now find myself packing a gat and running numbers.

Enough is enough. I intend to try to salvage something of my life. I am going back to work, giving up cheap cigars, AND, what is most important to you, I am actually going to pay for upcoming issues. I am enclosing a coupon and a check.

M. Leeper
13830 Cambridge No. 111
Southgate, MI 48195

P.S. If the Feds ask, you know nothing about me, understand?

Railroad Talk

Dear Editor:

Tom Korb's article in the September/October 1977 issue was sort of interesting; however, it might have been more useful and informative if it was more accurate and up-to-date.

For example, take the first paragraph which states: "Starting this year...". Well, this statement is not too accurate, as ACI has been in use since 1967, it's starting to begin its eleventh year...

Continuing, "all the nation's railroad cars ... will have one thing in common ... a 12x26 inch color coded information sign attached to their sides." Well, this is not really true either, as the Association of American Railroads has just voted to end the ACI program. The survey of AAR members overwhelmingly voted out the ACI system. As one might suspect, the reason was that the system did not work. At least, it failed to reach a useful reading/reliability rate. The AAR is expected to adopt the recommendations of its members and vote to remove the ACI labelling requirement for interchange equipment, within the next month or two. Efforts by vendors of the discredited equipment to stop the abandonment proceedings both in federal courts and before the ICC have ended in their being turned down. The ICC ruled that the use of ACI identification was solely a requirement set up by the railroads thru the AAR and did not fall under their jurisdiction.

The fact is, this OCR system was unable to perform in the environment that the railroads operate in. Operating 24 hours a day, seven days a week, trying to read ACI labels on cars in various states of dirtiness at all sorts of train speeds, the OCR just didn't prove reliable. Also, the rise of computer networks has proven to be more efficient in keeping track of freight-car movements.

As a matter of interest, the ACI label did not record owners of equipment, rather it recorded operators. There is a difference: a lot of equipment is leased under equipment trusts.

David J. Williams
5079 Blacksmith Dr.
Columbia, MD 21044

Shoestring Timeshare

Dear Editor:

I was surprised to see in *Creative Computing* a brief note entitled "Shoestring Timeshare" (Nov-Dec 1977) which I wrote some months ago for the *AEDS Monitor*. There must be many others who are interested in an inexpensive timeshare system, judging from the number of inquiries I have received concerning our system. We have made considerable progress since that article was written and your readers may be interested in an update.

We presently have two fully implemented, debugged operating versions. The first program, 5KTS, is designed to patch into Processor Technology BASIC/5 (software #2). This copywrited program simply loads above BASIC/5 and patches in the necessary changes to enter and exit that program, does all the "bookkeeping" chores required to keep track of user work areas and stacks, and initializes ports. There are some restrictions placed on the BASIC/5 features when timeshared:

1. The MEM command is disabled.
2. The P.T. video display module cannot be used.
3. The ARG and CALL functions are disabled.

All other features of the language are left intact. This program can be adapted to other languages, and to hardware configurations other than ours.

A second program, CHATS (Clairemont High Altair Time Share), may be similarly patched into MITS extended BASIC (version 3.2). This program also loads immediately above BASIC. The memory above CHATS is automatically divided into equal user spaces for 2 to 6 users. Of course as the number of users increases, the response time increases depending upon the tasks to which the system is assigned. Our system is primarily used in an interactive mode by students learning to program in BASIC and so time lag is not particularly disconcerting. If 3 or 4 of the terminals are simultaneously operating in loops (FOR/TO etc.) the delay becomes noticeable in print-out time on other terminals.

A third program, CHAOS (Clairemont High Altair Operating system), is planned for early 1978. This extremely flexible operating system provides for multiple languages that can be called up on demand (MITS, Disk BASIC 4.1, our own 8080 Assembly language, monitor and editor (META), a home-grown Pascal-like language and SHELL, a language unique to CHAOS that allows easy implementation of user-defined commands. This operating system resembles the UNIX system used at several branches of The University Of California.

We regularly operate our system on either an Altair 8800 or an Imsai or with minor modifications on a Sol computer. Other software and hardware requirements are as follows:

1. Altair SI/O serial ports (single or double), one per terminal.
2. A cassette interface (Tarbell)
3. A copy of Processor Technology BASIC/5 #2, or MITS BASIC (3,2) or MITS Disk BASIC (4.1) which are available from computer stores or directly from the producers of these languages.
4. RAM 16 K or more for 5 KTS; 24 K or more for CHATS; 48 K or more for CHAOS.
5. CHAOS requires at least one disk drive (MITS).
6. Cassette recorder/player; most student programs are stored on student owned cassettes.

Our plans for the future include use of the Altair as a buffer between the Imsai, disk drives, a Hewlett-Packard 2000 ACCETS system port and other peripherals such as a line printer and several recorder/players. We also expect to implement additional languages (APL, COBOL) to be included in the CHAOS.

If readers are interested in implementing our timeshare on their Intel-8080-based computer we hope soon to be able to provide a cassette tape and/or source listing for patching into the MITS extended BASIC (version 3.2). For further information contact the Clairemont High School Computer Club, Attn: R.D. Haas.

Robert D. Haas,
Club Sponsor
Clairemont High School
4150 Ute Dr.
San Diego, CA 92117

YES YOU CAN

Put The VERSATILE 2 To Work Now!

Even if you've NEVER used a computer, you can begin to operate the VERSATILE 2 immediately. It's all been done for you: a compact computer that needs no extras, and a complete software library.

Great for HOBBYISTS • TEACHERS • BUSINESSMEN

HOME You don't have to be a programmer or electronics technician. Or know anything more than how to plug it in, and read our manual. Get your hobby going without hassle.

CLASSROOM Get your kids right to work learning BASIC, playing games and solving math and accounting problems without worrying about what makes the computer go.

OFFICE Stop drowning in paperwork. And quit worrying about which system will work best for you. With our business accounting package, you can concentrate on business while the VERSATILE 2 concentrates on maintaining your records compactly, and permanently on diskette.

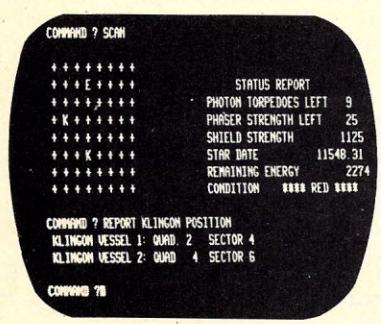
Shugart Mini-Floppy Disk Drive and Controller built into the VERSATILE 2

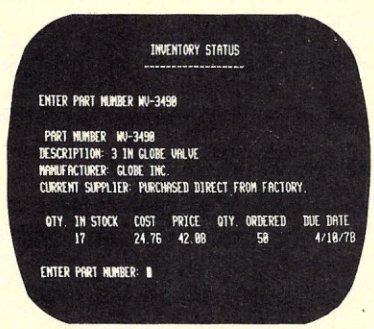
Numeric Keypad may be ordered as option at no charge.

SPECIFICATIONS
Z-80 CPU
Video display with graphics on 9" 64x16 screen
16K Static RAM Memory
Serial and Parallel I/O Ports with standard RS-232 Connector provided at rear of unit
ROM to drive RS-232

EXPANDABLE
Add memory, printer, and up to 3 external mini-floppy disk drives.

OPTIONS
8K Memory Boards at $195.00 each.





HOME SOFTWARE LIBRARY

Five diskettes are included to give you immediate programming capabilities.

DISK #1 contains a Disk Operating System and 12K Extended BASIC. Easy to use statements include: IF THEN, GOTO, READ, EXIT, FOR, NEXT. You get complete line editor, multi-statement lines and multi-dimension arrays. A BASIC teaching program on this diskette will have you quickly programming in BASIC.

DISK #2 has many games including STAR-TREK, BLACKJACK and STAR WARS. There's room left over for you to add your own.

DISK #3 is a Home Accounting Package with programs like Budgeting, and Checkbook Balancing.

DISK #4 contains a Small Business Accounting Package. Included are programs for Payroll, Inventory Control, Accounts Payable and Receivable, Taxes, Invoicing, Check Printing, and much more.

DISK #5 is a formatted blank diskette for you to enter your own programs.

VERSATILE 2

$2495 Assembled and Tested. 6 Month Warranty. 30-day delivery or available direct from dealers.

Scientific Sales, Inc.
175 W. Wieuca, Suite 210
Atlanta, GA 30342
(404) 252-6808

COMPUTER DATA SYSTEMS
5460 Fairmont Drive
Wilmington, Delaware 19808
(302) 738-0933

Alexander and Company, Inc.
5518 Florin Road
Sacramento, CA 95823
(916) 422-9070

See us at the Second West Coast Computer Faire – Booth #115

CIRCLE 153 ON READER SERVICE CARD

COMPLEAT COMPUTER CATALOGUE

We welcome entries from readers for the "Compleat Computer Catalogue" on any item related, even distantly, to computers. Please include the name of the item, a brief evaluative description, price, and complete source data. If it is an item you obtained over one year ago, please check with the source to make sure it is still available at the quoted price.

Send contributions to "The Compleat Computer Catalogue," *Creative Computing,* P.O. Box 789-M, Morristown, NJ 07960.

BOOKS AND BOOKLETS

COMPUTER BIBLIOGRAPHY

Over 300 items (computer information sources and publications) are listed in *Computer Science and Technology Publications,* a 35-page reference booklet compiled by the National Bureau of Standards' (NBS) Institute for Computer Sciences and Technology. Selected entries focus on computer-assisted manufacturing, fingerprint identification, typesetting, and the Bureau's computer-related activities and research.

For a free copy of booklet LP-84, which includes price and ordering information for all items, write: Institute of Computer Sciences and Technology, NBS, Administration, Room A209, Washington, DC 20234.

CIRCLE 167 ON READER SERVICE CARD

BASIC INTERDIALECT TRANSLATABILITY

BASIC Revisited, subtitled "An Update to *Interdialect Translatability of the BASIC Programming Language"* has been published by CONDUIT in their series of CONDUIT Guides known as Aids for Transfer of Instructional Computing. The 38-page booklet tells of the ANSI committee's attempts to develop standards for BASIC, how only a minimum number of statements are now included, and then "lays down a simple set of guidelines that should be set when a BASIC program is written in order to produce a transferable program." The study is based on twelve BASIC dialects, ranging from those used for small desktop minis to timeshared computers, and includes the BASICs for the: DEC PDP-10 and 11, Dartmouth DTSS, Hewlett-Packard HP2000, Honeywell MULTICS, IBM 5100 and V.S., Univac 1100 and Wang 2200. "(The twelve covered should be considered as an update to the forty eight covered in a previous report, Isaacs, 1974, available from CONDUIT.)" $2.50.

CONDUIT, P.O. Box 388, Iowa City, IA 52240.

CIRCLE 169 ON READER SERVICE CARD

VENDOR LITERATURE

CAMELOT'S 1978 CATALOG

Camelot Publishing Company has a new 24-page catalog detailing their computer books, materials, and teaching aids. Over 70 items are described for use by teachers, students, and computer users. Free

Camelot Publishing Co., P.O. Box 1357, Ormond Beach, FL 32074. (904) 672-5672.

CIRCLE 170 ON READER SERVICE CARD

TIMESHARING FOR THE MICROCOMPUTER

MicroAge, the systems marketing division of the Byte Shops of Arizona retail computer stores, has an 8-page brochure on the Alpha Micro multi-user multitasking timeshared microcomputer system. The Alpha Micro is an S-100 bus compatible software development system that features a timeshared operating system with full utilities, multi-user structured file system with password security, disk file management system for floppy or hard disks, AlphaBasic extended compiler and re-entrant runtime software, free-form text editor and text formatter, multiple pass macro assembler and 16-bit microprocessor with hardware floating point computation. Complete systems can be configured with multiple terminals and peripherals, including hard-disk systems with up to 1200 megabytes on-line. Disk access methods include sequential, random and indexed sequential. Other languages currently under development are LISP, APL, Forth, 8080 Cross Assembler, with Fortran, COBOL and RPG languages in the planning stages.

MicroAge, 803 N. Scottsdale Road, Tempe, AZ 85281. (602) 894-1193.

CIRCLE 171 ON READER SERVICE CARD

CROMEMCO CATALOG

Cromemco's 24-page color catalog describes the company's complete line of computer hardware and software. Cromemco's line of computers is based on the Zilog Z-80 microprocessor. The Z-2D computer features a built-in floppy disk drive, S-100 bus compatibility, and especially rugged design; the System Three is a sort of deluxe version of the Z-2D, with up to four floppy-disk drives, a line printer, and a CRT terminal, and seems aimed primarily at the business and educational markets. Cromemco also offers separate S-100 bus cards such as its 4-MHz Z-80 processor, A/D interface with joysticks, and the TV Dazzler color graphics interface. Catalog is $1.

Cromemco, Inc., 2400 Charleston Rd., Mountain View, CA 94043. (415) 964-7400.

CIRCLE 172 ON READER SERVICE CARD

MSI CATALOG

Midwest Scientific Instruments' new 20-page catalog is available without cost from the manufacturer. MSI offers a new 6800-based computer system which uses the same bus structure as the SWTPC 6800 computer, but has some extra features, such as a heavy-duty power supply, increased room for expansion in the computer, and the MSIBUG ROM monitor. Midwest also offers the FD-8, a full-sized floppy disk memory unit for the MSI 6800 or SWTPC 6800 computers. The catalog also contains many other in-

teresting 6800-compatible boards, including RAM and EPROM cards, extender cards, printers, and CRTs. A review of the MSI FD-8 will appear in a future issue of *Creative*.

Midwest Scientific Instruments, 220 West Cedar, Olathe, KS 66061. (913) 764-3273.

CIRCLE 173 ON READER SERVICE CARD

MUSIC CATALOG

Computer Music Compositions of the United States-1976, a catalog of computer music representing over 100 composers, is now available. $5.

Theodore Front, 155 North San Vicente Blvd., Beverly Hills, CA 90211.

CIRCLE 174 ON READER SERVICE CARD

ORGANIZATIONS

ACM SIG ON PERSONAL COMPUTING

The Association for Computing Machinery has chartered a new Special Interest Group on Personal Computing, SIGPC, which will be operated exclusively for educational and scientific purposes in the design and applications of computer systems for personal uses. This includes personal computer systems for home, clerical, small business, management and recreational uses. It also includes the technology of such systems in software and hardware, and emphasizes techniques appropriate to the integration of such tools as graphics, speed, data management, and music systems. Dr. Portia Isaacson, who chaired the 1977 National Computer Conference, has been appointed chairperson of SIGPC. Dr. Isaacson's immediate plans for SIGPC include appointment of other officers, publication of a quarterly newsletter, and holding SIGPC's first business meeting at ACM '77 in Seattle. To join SIGPC write to the Association for Computing Machinery, PO Box 12105, Church Street Station, New York, New York 10249. The dues (which include a subscription to the newsletter) are: $5.00/year for Members, associates, and student members of the ACM (please include ACM member number); $13.00/year for non-ACM members. A newsletter subscription without membership is $12.00/year.

For further information on SIGPC programs, contact Dr. Portia Isaacson, The Micro Store, 634 South Central Expressway, Richardson, TX 75080. (214) 231-1096.

CIRCLE 175 ON READER SERVICE CARD

COMPUTER RETAILERS' ASSOCIATION FORMED

The Computer Retailers' Association, a trade association of computer stores, has been formed with 24 founding members. The founding members include stores from across the United States and one Canadian store. The objective of the association is to provide services that individual stores cannot effectively provide themselves. Examples of such services include compiling industry statistics, working with manufacturers to improve the relationship between computer stores and manufacturers, arranging for group insurance, providing information about the computer store business to the financial community, and to encourage high standards among computer retailers. The specific objectives will be determined by the membership.

Portia Isaacson, The Micro Store, 634 S. Central Expressway, Richardson, TX 75080. (214) 231-1096.

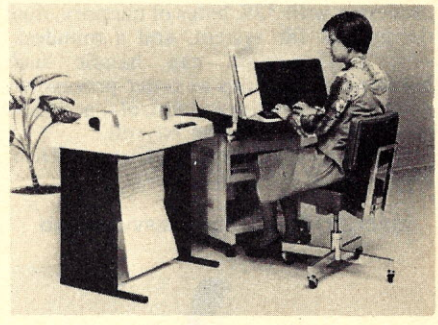
CIRCLE 176 ON READER SERVICE CARD

COMPUTERS

POLYMORPHIC KIT UPGRADES TO DISK

PolyMorphic Systems has introduced an upgrade kit for Poly 88 microcomputer owners who wish to convert their current systems to the company's new System 8813 disk-based microcomputer system. The kit contains all mechanical parts and electronic assemblies needed for converting a Poly 88, including chassis, walnut cabinet, a 10-slot backplane, power supply, floppy disk controller, 2K of read-only (ROM) memory, a fan, one floppy disk drive and two system diskettes. $1,450 from any PolyMorphic Systems dealer. Up to two more disk drives may be added at $590 each.

Toby Bradley, PolyMorphic Systems, Inc., 460 Ward Dr., Santa Barbara, CA 93111. (805) 967-0468.



LOW-END DEC DATASYSTEM

Newest and lowest-cost member of Digital's family of business computers, the DEC Datasystem 308 features a combined

terminal and processor, floppy-disk storage, and a choice of printers. Designed primarily for small businesses requiring only a single terminal and programming language, the Datasystem 308 operates from ordinary current and requires no special "computer room" environment. Data and program files for the new Datasystem are compatible with the larger Datasystem 310, and an optional word-processing program package is available for the 308. The 308 is compact and operates from ordinary current. A typical configuration consists of a video data processor with 32K bytes of memory, dual floppy-disk file system, and a minidesk. This configuration can handle such business applications as order processing, invoicing, inventory control, accounts receivable/payable, and payroll.

Digital Equipment Corporation, Maynard, MA 01754.

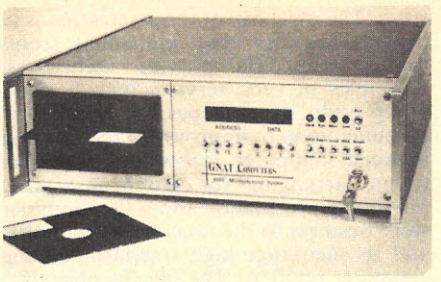
CIRCLE 178 ON READER SERVICE CARD

LOW-COST INTERACTIVE NCR COMPUTERS

A new NCR family of low-cost, interactive direct processing I-8130 and I-8150 systems that provide real-time data processing was described as "one of the most important product announcements by NCR in the past decade." Key aspects of the I-8100 family include low price, interactive direct processing (the ability to update files directly as data is entered into the system) and upward compatibility with larger members of NCR's 8000 computer series. Other features, including a large library of COBOL application packages, simplicity of operation, and self-instruction courses for owners and operators, are said to make the I-8100 family an appealing system for both small and large organizations. The new computers are designed as free-standing entry-level systems or for use in distributed processing networks. The I-8100 family will conform to NCR's recently announced Distributed Network Architecture. Primary markets for the system include manufacturing, wholesale, retail, government, financial, educational and health-care organizations. The basic version of the smaller member of the I-8100 family, the I-8130, sells for $19,970 and rents for $656 a month under a three-year agreement. It includes multiple processors, a 9-inch visual display, a keyboard, 48K bytes of memory, two flexible disk drives capable of storing up to 1 million bytes of information, and a bi-directional 50-line-per-minute matrix printer.

NCR Corp., Dayton, OH 45479.

CIRCLE 179 ON READER SERVICE CARD



FORTRAN ON MINI-FLOPPY/MICROCOMPUTER

A powerful Fortran compiler is now available with the Gnat minifloppy/microcomputer system. The Gnat Fortran System 8 operates the Fortran compiler under the Gnat CP/M Disk Operating System. The Fortran compiler includes all ANSI standard Fortran, X3.9-1966, except for double precision and complex numbers, which will be available at a later date. In addition to the Fortran compiler, a relocating assembler, a linking loader, and the Fortran library are included. The Fortran functions are implemented with 32-bit floating-point arithmetic. Integers are implemented with 16-bit numbers. The Gnat Fortran System 8 includes 1.3 microsec CPU, 32K RAM, 2K PROM on 16K ROM Module, serial/parallel I/O, disk interface and control, front panel, and minifloppy disk drive. Dual minifloppies and standard floppies are also available. Additional System Software includes Gnatbug monitor, disk operating system, assembler, editor and dynamic debugger. The system is completely assembled and tested and is ready to turn on and operate. $3990.

Frank Adams, Gnat Computers, 7895 Convoy Court, Unit 6, San Diego, CA 92111. (714) 560-0433.

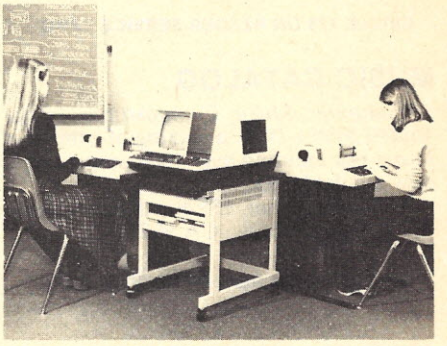
CIRCLE 180 ON READER SERVICE CARD

HP DESKTOP COMPUTER

The first in a new generation of desktop computer systems has been placed on the market by Hewlett-Packard. The new Series 9800 System 45 is said to have the most powerful central processor and the largest built-in mass storage system ever offered in a desktop computer. It also features a 12-inch CRT display, BASIC interpretive language conforming to the new ANSI standard, applications software, and an optional graphics package with high-speed hard-copy output. The system is all contained within a single compact package. The graphics mode provides a 560 x 455 dot matrix with high visual resolution and no perceptible flicker. The alphanumeric mode offers a full 80-character wide, 24-line deep screen.

Hewlett-Packard Co., 1507 Page Mill Road, Palo Alto, CA 94304.

CIRCLE 181 ON READER SERVICE CARD



LOW-COST EDUCATIONAL TIMESHARED COMPUTER

Claimed to be the lowest cost educational computer system available from a major manufacturer, the ECS-7800 from the Educomp Division of Quodata Corp. is $9,900.

The ECS-7800 was designed specifically for the educational marketplace and is based on the latest-technology member of Digital Equipment Corporation's popular PDP-8 family. It is a complete timesharing system with 32K bytes of MOS memory, two terminals (video display unit and DECwriter II), dual diskette storage, portable mini-cabinet and a sophisticated multi-user BASIC. Fortran IV and assembly language are included at no additional cost. ECS-7800 may be expanded to 16 users.

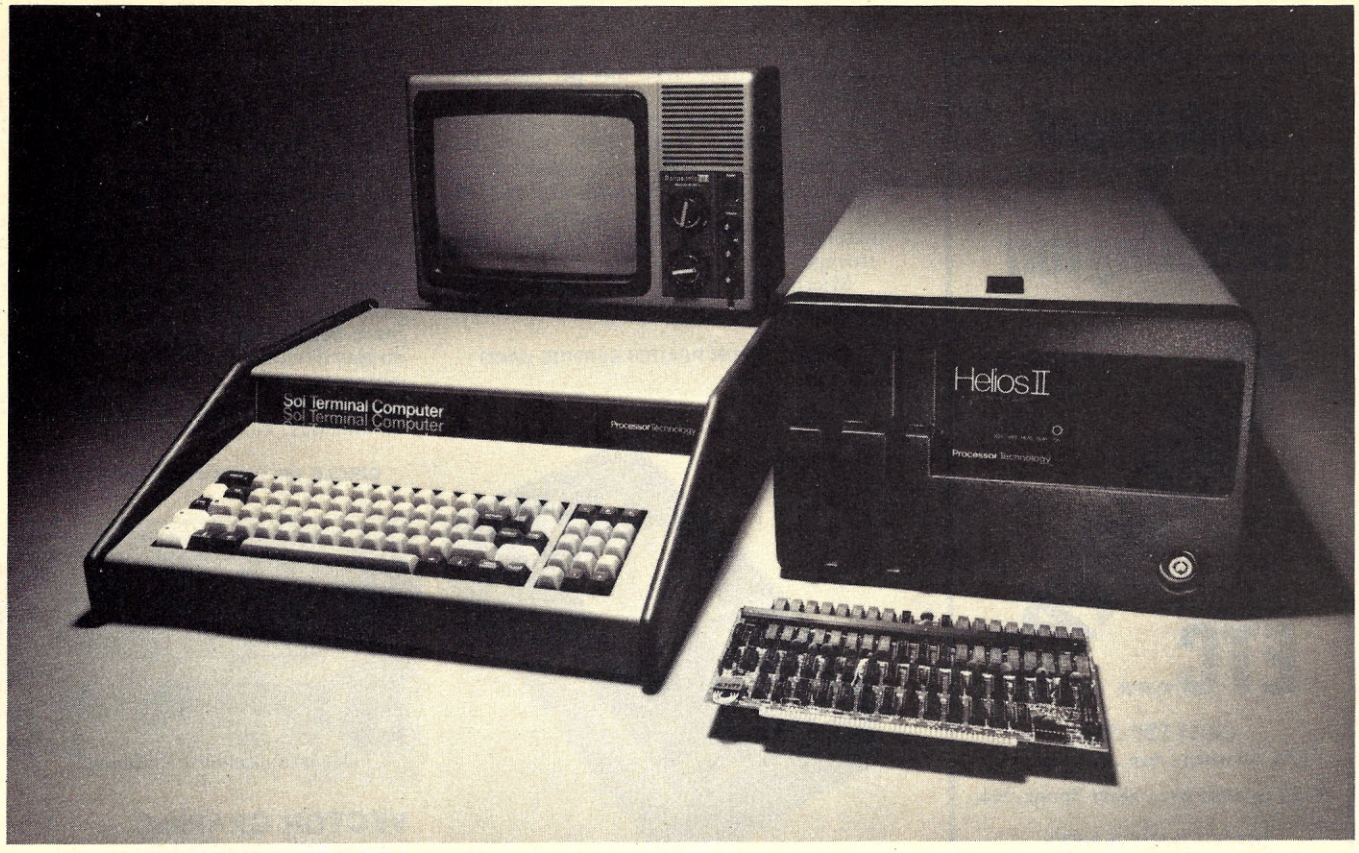
Educomp Div., Quodata Corp., 196 Trumbull St., Hartford, CT 06103. (203) 728-6777.

CIRCLE 182 ON READER SERVICE CARD

TERMINALS

MICON TERMINAL

A new, low-cost, flexible data communications terminal called the Miget is available from Micon Industries. Miget (Miniature Interface General-purpose Economy Terminal) provides keyboard entry and display output compatible with



Sol-20. First it was THE SMALL COMPUTER. Now, it's THE SMALL COMPUTER SYSTEM.

A year ago, we introduced the Sol-20. It wasn't the first small computer. It was the first complete small computer with everything needed to get it up and on the air as it came from the factory. The keyboard, interfaces, extra memory, factory backup, and service notes were all there.

The results are in: Sol-20 is now the number one small computer in the world. Sols aren't the cheapest, just the most valuable.

We originally designed the Sol-20 as the heart of a complete computer system. So now to solve the problems of science, engineering, education, business management and control and manufacturing, we offer fixed price Sol systems in either kit or fully tested and assembled form. We offer language flexibility, Extended BASIC, Assembler, PILOT* and FORTRAN* We offer Helios II/PTDOS, an extraordinarily capable disk operating system. And remember, though we call these small or personal computer systems, they have more power per dollar than anything ever offered. They provide performance fully comparable and often superior to mini-computer systems costing tens of thousands of dollars more.

What you get. What it costs.

Typical systems include Sol System I priced at $1600 in kit form, $2095 fully assembled and tested. Included are a Sol-20/8 with SOLOS personality module storing essential system software, an 8192 word memory, a 12" TV/video monitor, and a cassette recorder with BASIC tape.

Sol System II has the same equipment with a larger capacity 16,384 word memory. It sells for $1825 in kit form; $2250 fully assembled.

For even more demanding tasks, Sol System III features Sol-20/16 with SOLOS, 32,768 words of memory, the video monitor and the dual drive Helios II Disk Memory System with the PTDOS disk operating system and Extended DISK BASIC Diskette. Price, $5795 fully assembled and tested.

More information.

For the most recent literature and a demonstration, see your dealer listed below. Or if more convenient, contact us directly. Please address Processor Technology Corporation, Box O, 7100 Johnson Industrial Drive, Pleasanton, CA 94566. Phone (415) 829-2600.

*Available soon.

ProcessorTechnology

CIRCLE 107 ON READER SERVICE CARD

AZ: Tempe (602)894-1129; Phoenix (602)942-7300; Tucson (602)327-4579. CA: Berkeley (415)845-6366; Costa Mesa (714)646-0221; Fresno (209)266-9566; Hayward (415)537-2983; Lawndale (213)371-2421; Orange (714)633-1222; Pasadena (213)684-3311; Sacramento (916)443-4944; San Francisco (415)431-0640, (415)421-8686; San Jose (408)377-4685, (408)226-8383; San Rafael (415)457-9311; Santa Clara (408)249-4221; Sunnyvale (408)735-7480; Tarzana (213)343-3919; Van Nuys (213) 786-7411; Walnut Creek (415)933-6252; Westminster (714)894-9131. CO: Boulder (303)449-6233; Englewood (303)761-6232. FL: Fort Lauderdale (305)561-2983; Miami (305)264-2983; Tampa (813)879-4301. GA: Atlanta (404)455-0647. IL: Champaign (217)359-5883; Evanston (312)328-6800; Lombard (312)620-5808. IN: Bloomington (812)334-3607; Indianapolis (317)842-2983, (317)251-3139. IA: Davenport (319)386-3330. KY: Louisville (502)456-5242. MI: Ann Arbor (313)995-7616; Royal Oak (313)576-0900; Troy (313)362-0022. MN: Minneapolis (612)927-5601. NJ: Hoboken (201)420-1644; Iselin (201)283-0600. NY: Middle Island (516)732-4446; New York City (212)686-7923; White Plains (914)949-3282. NC: Raleigh (919)781-0003. OH: Columbus (614)486-7761; Dayton (513)296-1248. OR: Beaverton (503)644-2686; Eugene (503)484-1040; Portland (503)223-3496. RI: Warwick (401)738-4477. SC: Columbia (803)771-7824. TN: Kingsport (615)245-8081. TX: Arlington (817)469-1502; Houston (713)526-3456, (713)772-5257; Lubbock (806)797-1468; Richardson (214)231-1096. VA: McLean (703)821-8333; Reston (703)471-9330; Virginia Beach (804)340-1977. WA: Bellevue (206)746-0651; Seattle (206)524-4101. WI: Milwaukee (414)259-9140. WASHINGTON D.C.: (203)362-2127. CANADA: Ottawa (613)236-7767; Toronto (416)484-9708, (416)482-8080, (416)598-0262; Vancouver (604)736-7474, (604)438-3282.

CIRCLE 154 ON READER SERVICE CARD

all microcomputer and microprocessor systems using RS-232C interface and ASCII code. Features of the four-pound Miget include eight selectable baud rates from 110 to 9600, complete TTY compatibility, an optional self-contained memory system and acoustic coupler, and a choice of eight colors. $400.

Micon Industries, 252 Oak Street, Oakland, CA 94607. (415) 763-6033.

CIRCLE 183 ON READER SERVICE CARD

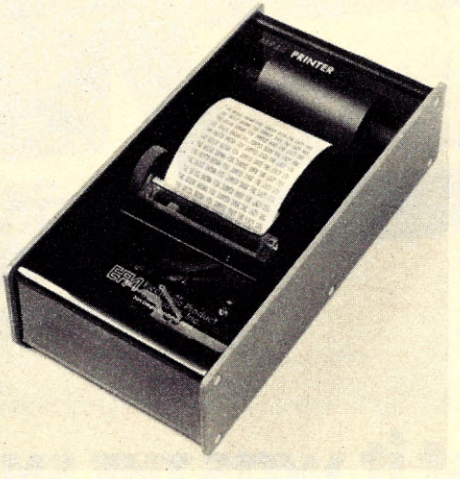
PERIPHERALS

MACROFLOPPY

A new series of fully packaged and assembled disk drives priced as low as $695 — including software, S-100 compatible controller and 143,000-byte capacity — was announced by Micropolis Corporation. Model 1041 includes a drive, enclosure, cabling and connectors, disk operating system and disk extended BASIC at a suggested retail price of $695. It is intended for integration into any 8080A or Z-80 microcomputer chassis. Model 1042, suggested retail of $795, adds a power supply and DC regulators for desktop use. MacroFloppys are hard-sectored into 16 sectors, each 256 bytes long; total tracks per surface is 35. Both offer transfer rates of 250K bits/second at an average rotational latency time of 100 msec. Access time track to track is 30 msec, and recording density 5162 bits per inch. The disk operating system includes assembler, file management routines and utilities to support 8080A and Z-80 programs. MacroFloppy DOS will function in microcomputer systems with at least 16K of main memory. The company's disk extended BASIC requires 24K of main storage.

Micropolis Corp., 7959 Deering Ave., Canoga Park, CA 91304. (213) 703-1121.

CIRCLE 184 ON READER SERVICE CARD



EPA MINIPRINTER

The Electronic Product Associates MP-44 Mini Printer is a simple and inexpensive 5 x 7 dot-matrix printer for microcomputer systems. Electrosensitive paper is used to make permanent copies at speeds up to 88 characters per second with 44 characters per line. Software control allows expansion of character size for emphasis. Black characters are printed on aluminized paper 6 centimeters (2 3/8 inches) wide. An enclosure 4 1/2 by 8 3/4 inches houses the printer and paper supply mounted on an interface board with all necessary components for connection to any microcomputer with TTL logic levels. $257.

Chuck Bennett, Electronic Product Associates, Inc., 1157 Vega St., San Diego, CA 92110. (714) 276-8911.

CIRCLE 185 ON READER SERVICE CARD

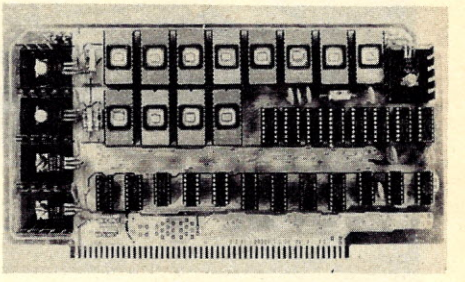
MISC. HARDWARE

PROGRAMMABLE CHARACTER GENERATOR

Objective Design announces the programmable character generator for S-100 Computers. This new S-100 card adds the ability to dynamically create the characters generated by a video display device. For those who require special mathematical or scientific symbols, APL characters, sub- and super-scripts, high-density bar graphs, Greek letters, or game characters such as space ships, the programmable character generator allows the creation and storage of the new characters while retaining intact the original character set. The original character set remains available for use at any time. Keyboard interface and dual joystick interfaces are provided on the board. The programmable character generator is an ideal addition to Sol terminals, the Polymorphic VTI, the Processor Technology VDM-1, the Solid State Music video board, and other video display devices utilizing the Motorola 9 x 7 matrix character generator.

Objective Design, Inc., P.O. Box 20325, Tallahassee, FL 32304.

CIRCLE 186 ON READER SERVICE CARD

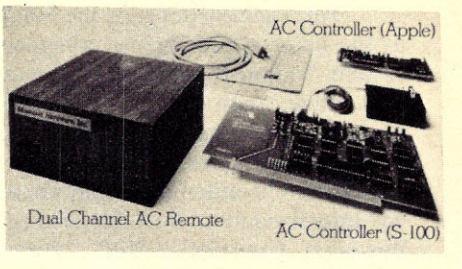


VECTOR GRAPHIC PROM/RAM BOARD

Vector Graphic is introducing a new PROM/RAM board with 1K on-board RAM and capacity for up to 12K 2708-type EPROMs. The board occupies two independently addressable 8K blocks. Complete addressing flexibility is provided to conform to virtually any system configuration with a minimum of address jumpers required. Video boards or disk operating systems can be nested in the 3K of unused space. MWRITE logic and jump-on-reset allow operation without a front panel. A 24-command PROM monitor is available to interface with most popular I/O boards. $135 kit, $175 assembled.

Contact your local dealer or Vector Graphic Inc., 790 Hampshire Road A-B, Westlake Village, CA 91361.

CIRCLE 187 ON READER SERVICE CARD



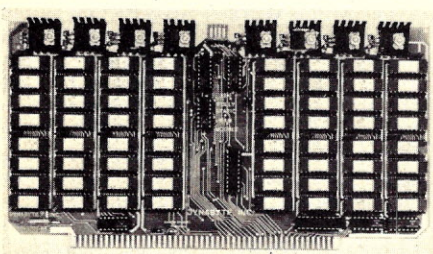
COMPUTER CONTROL THRU AC WIRING

A new system designed to simply and economically control AC devices remotely from any S-100 bus or Apple II computer over existing 110-VAC wiring in homes, factories, schools and businesses has been announced by Mountain Hardware, Inc. Designated Introl, the new unit provides

on/off control and status checks at any AC outlet. The system impresses a code-modulated 50-KHz control on the ordinary AC wiring, then decodes it at any outlet to switch AC devices on and off. In the home such devices could include lights, TV's, stereos, solenoid valves, sprinklers, burglar alarms and so on. With the addition of input sensors, the computer system can control automatically such variables as temperature, humidity and soil moisture. Similarly in office, school and factory environments, the system can manage energy, heating, ventilation, air conditioning, security, processes and almost any other series of sequential events. A single AC controller board plugs into the computer bus, connects to the AC Interface adapter plugged into any convenient 110-VAC outlet, and can address as many as 64 channels remotely. Programs are written in BASIC or Assembler language. Software subroutines come with the equipment. For S-100 computers, a 100,000 day Calendar/Clock Board is an option. AC Controller, $149 kit, $189 assembled. Dual Channel AC Remotes, $99 kit, $149 assembled. Calendar/Clock Board, $179 kit, $219 assembled.

Mountain Hardware, Inc., P.O. Box 1133, Ben Lomond, CA 95005. (408) 336-2495.

CIRCLE 188 ON READER SERVICE CARD

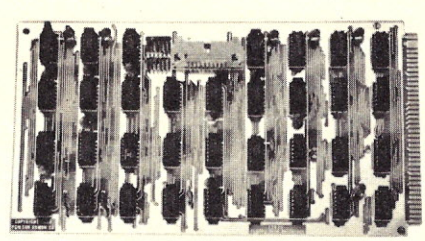


16K, 32K STATIC RAMS

Two fully static RAM modules for the S-100 bus are now available from Dynabyte. The 16K static RAM and the 32K (shown) static RAM are available with access times of either 450 or 250 nsec; the latter is compatible with 4MHz Z-80 processors. The new RAM modules' fully static functioning and their complete buffering make them completely compatible with all known processors, including the Z-80s offered by several manufacturers as well as the Alpha Microsystem A-100. Both 16K RAM modules feature Bank Select, which allows up to eight separate banks (of up to 64K each) to reside in the same system. The module may be addressed in four separate 4K blocks along 4K boundaries. Each of these 4K blocks may be individually Write Protected. If an attempt is made to write into a protected block, an audible alarm will be activated and a visual indicator will be displayed for several seconds. The 32K static RAM modules offer 4K boundary addressing, complete buffering, and conservative thermal design. 1645 (16K, 450 nsec.) $525; 1625 (16K, 250 nsec.) $555; 3245 (32K, 450 nsec.) $925; 3225 (32K, 250 nsec.) $995.

Dynabyte, Inc., 4020 Fabian, Palo Alto, CA 94303. (415) 494-7817.

CIRCLE 189 ON READER SERVICE CARD

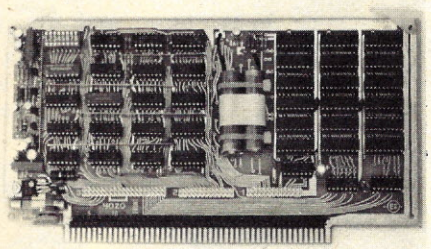


DUAL FLOPPY-DISK INTERFACE

Pacific Cyber/Metrix, has announced a dual floppy-disk interface module for its 12-bit microcomputer system, the PCM-12, which is software-compatible with Digital Equipment Corp.'s PDP-8 family of minicomputers. PCM's floppy-disk module interfaces the PCM-12 to Data Systems Design's 210 floppy-disk memory system. Fully plug-compatible with the 210, PCM's 12440 floppy-disk module will allow PCM-12 users to execute all PDP-8 floppy-disk diagnostics and makes the PCM-12 system fully compatible with all mass-storage operating systems already developed for the PDP-8 family. $259 assembled, $169 kit.

Ted Netoff, PCM, 3120 Crow Canyon Road, San Ramon, CA 95483. (415) 837-5400.

CIRCLE 190 ON READER SERVICE CARD



NON-VOLATILE RAM

Electricom's 4020 non-volatile high-speed semiconductor RAM memory has a Size and word width of 2Kx8/9 or 1Kx16/18, jumper-selectable. The 4020 is a single card memory designed to meet the requirements for short and long-term non-volatile high-speed Random Access Memory systems. Memory data is maintained for a minimum of three months (six months typical) after the primary board power is removed. The 5 x 10 inch card features a 450ns access time, bank select within 64K, S-100 data bus compatability, and LS type TTL interface. On-board nickel-cadmium batteries, battery charger and power-state monitors eliminate the need for external support circuitry. $287.

Pat Patterson, ElectriCom Co., 12567 Crenshaw Blvd., Hawthorne, CA 90250. (213) 676-6576.

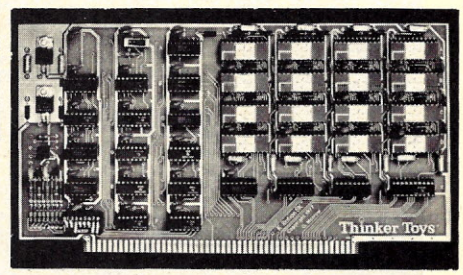
CIRCLE 191 ON READER SERVICE CARD

VIDEO DISPLAY SYSTEM

The MSDV-100 Video Display System is a high-quality 80-character, 24-line video output device for the S-100 bus. The character set includes upper and lower case characters as well as full punctuation. Any character can be underlined, a feature useful in word processing. A character can also be made to blink at a user selectable rate, often used for alarm or warning situations. Additionally, a character can be made to appear brighter than normal or to appear in a reverse field (black on white), useful in order entry or other applications to highlight text. The MSDV-100 can generate high-quality forms overlays. Margins can be either single or double wide with continuous intersections. Charts, graphs, or order entry forms are easy to produce on the video screen. The MSDV-100 displays continuous grey-scale elements in any of nine levels in any of 1920 positions on the screen. This is especially useful for bar graphs and for grey-scale graphics or animations, as well as in forms applications. The MSDV-100 is a two-board S-100 based system which occupies 2K of RAM address space and two Input/Output ports. $285 kit.

Micro Systems Development, 2765 So. Colorado Blvd., Denver, CO 80222. (303) 758-7411.

CIRCLE 192 ON READER SERVICE CARD



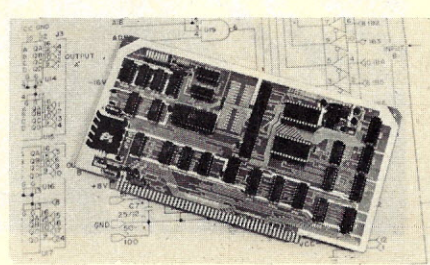
DYNAMIC MEMORY

A new concept of memory refreshing circuitry for S-100 micros, named SynchroFresh, is said to be much simpler than previous approaches and totally reliable. Using SynchroFresh, the new 8K memories use half the power of static boards, and "can undersell both static and older design dynamic memories by considerable margin." The refreshing system eliminated reliability problems because it does not interrupt normal CPU operations or timing in order to perform memory refresh. SynchroFresh circuitry simply monitors the microprocessor's machine states, utilizing the T4 states for refresh. T4 always occurs during instruction fetches, leaving memory available for refresh.

The first dynamic memory equipped with SynchroFresh, Econoram III 8Kx8, costs $149 assembled.

Thinker Toys, 1201 10th St., Berkeley, CA 94710. (415) 527-7548.

CIRCLE 193 ON READER SERVICE CARD

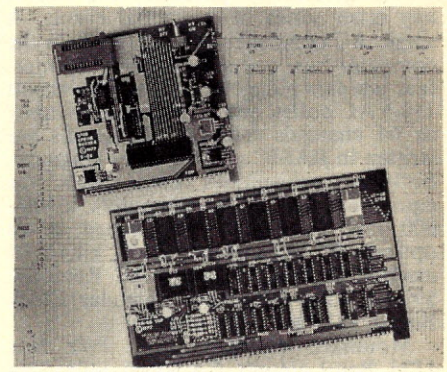


"BIT STREAMER" I/O BOARD

A "Bit Streamer" I/O board, available assembled or in kit form from Vector Graphic Inc., combines two parallel input and output ports, and a serial I/O port using an 8251 programmable universal synchronous/asynchronous receiver-transmitter. Communications with board circuitry is accomplished by the CPU. One parallel port also can be used as a keyboard input port. The USART is designed to interface easily to an S-100 bus structure and is capable of being configured for a wide variety of communication formats. The "Bit Streamer" has been designed for ease of construction. Without introducing changes to the pre-jumpered options, the board can be installed on a computer and will operate as an RS232 serial port using the initialization and I/O software on the Vector Graphic option C PROM. Available from Vector Graphic computer-store dealers, the board is $155 kit, $195 assembled.

For technical data: Vector Graphic Inc., 790 Hampshire Rd., Westlake Village, CA 91361. (805) 497-6853.

CIRCLE 194 ON READER SERVICE CARD



PROM BOARD AND EPROM PROGRAMMER

The Micro Works PSB-8 PROM system board features 1K of high-speed (350 ns), low-power RAM and space for up to eight 2708 EPROMs, both DIP-switch addressable to start on any 8K boundary in memory. The exclusive I/O select feature allows the user to move the I/O locations, up to any unused 1K block in the EPROM memory space, permitting memory expansion to a full 56K of contiguous user RAM. PSB-08 (EPROMs not included), $119.95; PSB-08R (regulated +12) $124.95.

The B-08 is a compact 2708 EPROM programmer that fits in a standard SWTPC 6800 I/O slot. A safety switch and LED indicator provide control over the high programming voltage generated on the board. A zero-insertion-force socket and extended board height allow effortless PROM insertion and retrieval. Fully commented source listings of the Micro Works U2708 PROM Utility software are included, allowing quick and reliable programming and copying of 2708s. An optional + 12 volt regulator is available. B-08, $99.95; B-08R (regulated +12), $104.95.

The Micro Works, P.O. Box 1110 DelMar, CA 92014. (714) 756-2687.

CIRCLE 195 ON READER SERVICE CARD

SOFTWARE

CROMENCO 16K BASIC

Cromemco has announced the 16K Extended BASIC, with disk file and line-printer I/O. The BASIC "pre-interprets" program lines as they are entered, thus catching many syntax errors before they become part of the program. BCD arithmetic is used to prevent round-off errors. Cromemco 16K BASIC also has a trace feature, error trapping, and a com-

plete set of BASIC statements and functions, including a MAT function that initializes all the elements of an array to a specified value. FOR loops are automatically indented, and an ON ESC command transfers control to any line in a program when the Escape key is pressed. On soft-sectored IBM compatible diskettes, $95.

Cromemco, 2400 Charleston Rd., Mountain View, CA 94043. (415) 964-7400.

CIRCLE 196 ON READER SERVICE CARD

OSI WORD PROCESSOR

Ohio Scientific announces its new Word Processor OS-WP1, a full text editor that operates at both the character and line levels. It has its own internal GET and PUT file commands which transfer individual files from memory to disk by typing the appropriate command. A total of 209,000 characters can be stored on a diskette. The OS-WP1 has a full set of printer-control commands that can be used with virtually any impact or matrix computer printer or word-processing printer. The formatted output mode allows the user to perform left and right justification of text without line numbers at a designated width of from 20 to 70 characters. The OS-WP1 is for writing letters, manuals, reports, and all normal everyday business forms. It can be used directly with the Lear Siegler ADM-3A or with the Hazeltine 1500 and is adaptable to virtually any other conventional CRT terminal via documentation provided. The complete Word Processor package, two diskettes and a manual is now available for only $79 for use on any disk-based Ohio Scientific computer system.

Ohio Scientific, Hiram, OH 44234. (216) 569-7905.

CIRCLE 197 ON READER SERVICE CARD

EMPL/8080 INTERPRETER

Have you exhausted BASIC? Are you looking for a new and interesting language for your microcomputer? EMPL is an easy-to-learn micro version of APL for the Intel 8080. It resides in the first 5632 bytes of memory. EMPL has numeric and character vectors, user-defined niladic, monadic and dyadic functions, 22 primitive functions, 9 system commands, and many other special operators and characters. EMPL can be run either in the ASCII or APL character set. Double-byte integer arithmetic is used. EMPL comes with a user's manual that includes complete information on implementing it on any Z-80/8080 system with at least 8K of memory. $10 on Tarbell cassette; $20 on paper tape, North Star disk, CUTS cassette, or MITS cassettes.

Erik T. Mueller, Britton House, Roosevelt, NJ 08555. (609) 448-2605.

CIRCLE 198 ON READER SERVICE CARD

TEXT EDITOR

Interactive Information Systems has released a new software product, "Page," which will display any specified text on the screen of the CRT terminal. The user can then visually locate any changes that need to be made and key in the change. The change will be immediately reflected on the screen. Anyone presently using Digital Equipment Corp. computers operating under RSTS/E can implement Page on their system. $750.

Interactive Information Systems, Inc., 10 Knollcrest Dr., Cincinnati, OH 45237.

CIRCLE 199 ON READER SERVICE CARD

THE DEVIL'S DUNGEON

Engel Enterprises' second publication for the computer hobbyist is based on the game, "Dungeons and Dragons." As in the game, the player will get lost in the bottomless dungeon, haunted by monsters, demons, volcanic tremors, and poisonous gas, without a map. This program, written by Dr. C. William Engel in MITS - 8K-3.2 BASIC, is documented in the same manner as his first book, *Stimulating Simulations*, with a scenario, sample run, flowchart, list of variables, program listing (108 lines), and suggestions for expansion and modification. $3.50.

Engel Enterprises, P.O. Box 16612, Tampa, FL 33687.

CIRCLE 200 ON READER SERVICE CARD

INTERPRETIVE DEBUGGING FOR THE 8080

DBUG: An 8080 Interpretive Debugger, a program for entering, debugging and storing assembly language programs, is now available in a book from E & L Instruments, Inc. The 100-page paperback is the first of the Bugbook Application Series on assembly-language programming. DBUG permits the user to enter a program into an 8080 microcomputer memory and single-step it through, instruction by instruction. DBUG is an aid for those who develop 8080 microcomputer software. With DBUG the user can enter and change data and program steps stored in random-access memory. After a program is entered, it can be single-stepped by using the break point to observe the effect of a particular instruction on each of the 8080's internal registers. DBUG steps through one complete instruction regardless of the number of cycles required. DBUG will reside in a 1K byte block of memory and a bootstrap loader for loading the DBUG into memory is included. Two complete listings of DBUG are given in the appendixes—one in octal code and the other in hexadecimal code, each with appropriate I/O subroutines. $5.

E & L Instruments, Inc., 61 First St., Derby, CT 06418. (203) 735-8774.

CIRCLE 201 ON READER SERVICE CARD

Z-TEL TEXT-EDITING LANGUAGE

Technical Design Labs announces the new Z-TEL Text Editor; a Z80 Text-Editing Language. Z-TEL is a utility program designed to provide a powerful set of techniques for editing and manipulating text files. Z-TEL has the capability of easily moving large blocks of text inside the buffer. The user can avoid the deletion and manual retyping of text. The macro capabilities of Z-TEL provides the option of tailoring the program to the user's individual needs. A macro expression is a string of commands stored in one or more of Z-TEL's ten text registers. When a macro command is typed, Z-TEL will execute the series of commands using specified text register(s) and eliminate the repeated typing of similar or identical commands. Z-TEL includes detailed error detection. Z-TEL is a relocatable, ROMable program which requires less than 7K of memory. $50 on paper tape, $40 on cassettes, will be available on disk.

Donna Galletti, Director of Sales, Technical Design Labs, 1101 State Road, Princeton, NJ 08540. (609) 921-0321.

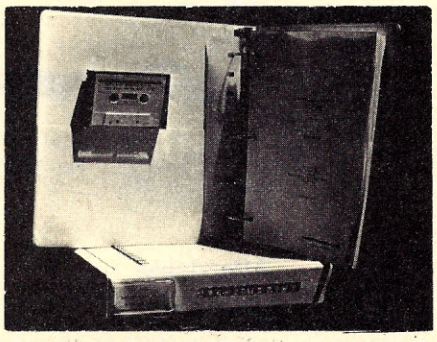
CIRCLE 202 ON READER SERVICE CARD

SOFTWARE DYNAMICS BASIC COMPILER

Use it to build business applications. That's what it was designed for. Decimal arithmetic for penny amounts up to $99,999,999.99. Formatted output, strings and file I/O are included. Long variable names really aid program documentation. With the compiler you can sell programs written in BASIC without revealing the source code. Got some heavy computation? Fast software floating point gives you a boost in speed: The compiled nature of SD BASIC gives you a boost, and automatic integer optimization on arithmetic, for/next loops and subscripting gives you a third boost. Process control? SD BASIC's speed, floating point, peek/poke and I/D allows many control programs to be built in something other than assembly language. Program generation for SD BASIC is done using whatever text-processing systems is available. The compiler processes this text (from cassette, floppy disk, etc.) and produces an intermediate file which is assembled using the SD assembler.

Smoke Signal Broadcasting, Box 2017, Hollywood, CA 90028. (213) 462-5652.

CIRCLE 203 ON READER SERVICE CARD



ZAPS CASSETTE OPERATING SYSTEM

The ZAPS Cassette Operating System consists of a full Z-80 assembler, text editor, in-memory file system, labeled cassette-tape storage system with full cyclic redundancy checks on all tape operations, and a variety of other software utilities. The system is entirely resident in 14K of memory including all buffers and 1K bytes of symbol table space. It will run on most 8080 and Z-80 processors. It is distributed on Tarbell, Digital Group, and TDL standard tapes. The ZAPS text editor is much like those used on time-sharing systems. Because it is a full-context editor, no line numbers are required. Only the actual data bytes are stored. Editor commands include input, insert, delete, replace, change, global change, up, down, top, bottom, string search, print and print current line number. The system comes with a 9-page user manual. Sections of the manual include examples for all commands, debug utilities, and listings of all input/output drivers. $60.

Algorithmics Inc., Box 56, Newton Falls, MA 02164. (617) 695-0545.

CIRCLE 204 ON READER SERVICE CARD

MISCELLANEOUS

COMPUTER-PROJECT KIT

Edu-Pak Publishing has introduced a seven-piece computer-project kit for students in grades 4 through 9. The kit provides a brief history of the computer, tells how to get data into a computer, and answers a list of most frequently asked questions. Included in the kit is a prepunched tabulating card, a mark-sense card, a printed Teletype-tape decoder sheet, and samples of magnetic and Teletype tape. From 20 cents each for 50, to 11 cents each for 1,000.

Edu-Pak Publishing, Box 27101, Minneapolis, MN 55427.

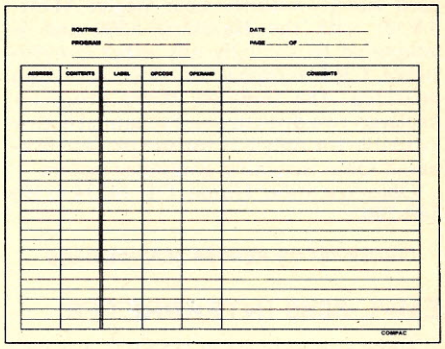
CIRCLE 205 ON READER SERVICE CARD

MAIL-ORDER MAGNETIC MEDIA

Printcraft Systems has formed a mail-order division to handle small-quantity orders for diskettes, cassettes, ribbons, print wheels, etc., for the home computer hobbyist. The same product line being offered their commercial accounts, such as 3M Scotch Brand, ITC Verbatim, etc., will be made available to the hobbyist, in less than standard packaging. The division was formed to assist hobbyists in being able to enjoy their hobby at a reasonable cost. They can charge their purchase to their Master Charge and Bank Americard (Visa) accounts.

Printcraft Systems, Inc., Dept. MO, 11-17 Beach St. (off Canal St.), New York, NY 10013.

CIRCLE 206 ON READER SERVICE CARD



CODING FORM, CONVERSION TABLE

Compac offers, as an aid for assembly-language programming, coding forms said to organize and simplify your source code. "Specifically designed to be compatible with all of the most popular microprocessor assembly language formats," the forms have room for labels, opcodes, operands, and line-by-line documentation, plus space for object code and address, if you don't have your own assembler and printer. At 50 sheets per pad, 25 lines per page, predrilled to fit a three-ring binder, the pads are $2.25 each, $6. for three.

The two-sided one-page base-conversion table allows the programmer to look up the representation of a number in a different base. The table contains the numbers 0 through 255 tabulated in five different number systems: binary, octal, decimal, hex, and two's complement. $2 each, three for $5.

COMPAC, P.O. Box 18470, Cleveland, OH 44118.

CIRCLE 207 ON READER SERVICE CARD

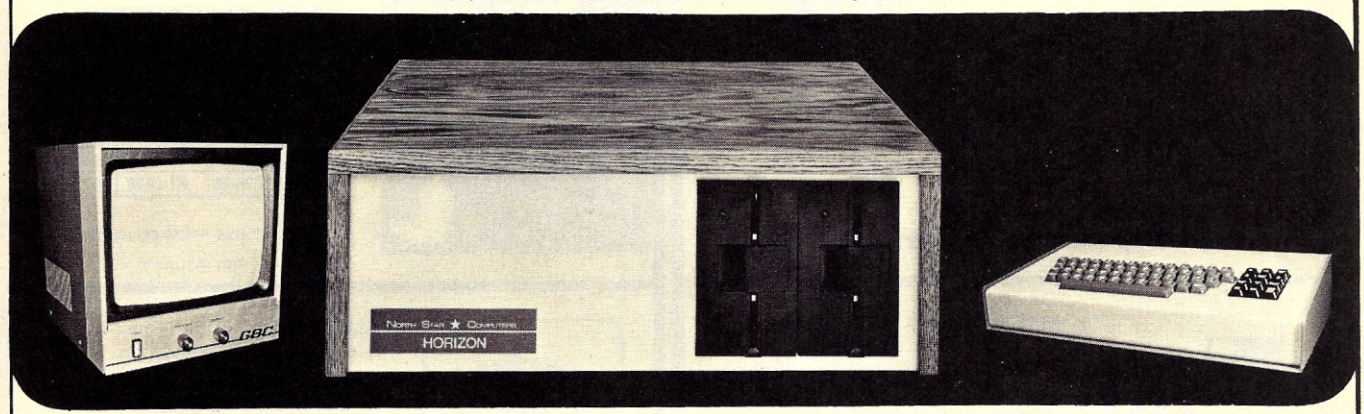
ASSEMBLED SYSTEMS
With Disk Capability
AT KIT PRICES!

ISN'T YOUR TIME WORTH $58.00?

Then why spend needless time and energy when we will deliver assembled and fully tested systems, like this one.

Ideal for the BUSINESS OFFICE or the CLASSROOM

North Star HORIZON



North Star Horizon Single Drive System includes the Z-80 CPU at 2 or 4 MHz, motherboard, 16K of memory at 4 MHz and power supply. Software includes Disk Operating System and Disk BASIC. Horizon 1 kit is $1599. Dual Drive Horizon is also available at $1999.

We add monitor and keyboard.

Compare our assembled prices and save hours of soldering, testing and trouble shooting!

Here is what you would pay if you bought these components as separate kits.

Component
North Star HORIZON 1
Parallel Input/Output
PROM
Video Board (64 by 16) ✱
9" Video Monitor
ASCII Keyboard and Enclosure

Your cost for separate kits would total $2238.00.

Your assembled price from Sunshine Computer Company is $2296.00.

OPTIONS

✱Move up to a **Hazeltine 1500 CRT Terminal** for an additional $595.00.

Dual Drive $395.00

SYSTEM SOFTWARE GIVES YOU TRUE DISK FILE CAPABILITY

You get the Horizon 1 complete with North Star Disk BASIC. A complete business package on diskette is available for $295, and includes:

- General Ledger
- Accounts Receivable
- Accounts Payable
- Payroll
- Inventory
- Amortization
- Mailing List

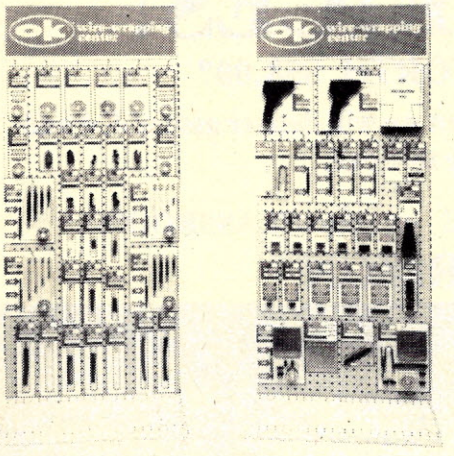
Assembled systems sold with 90-day written warranty. Come in and see our Horizon in operation.

Sunshine Computer Company

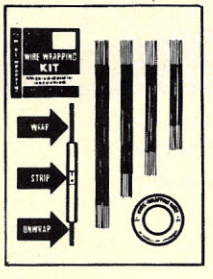
20710 South Leapwood Ave. • Carson, California 90746 • (213) 327-2118

CIRCLE 165 ON READER SERVICE CARD

ok® wire wrapping center



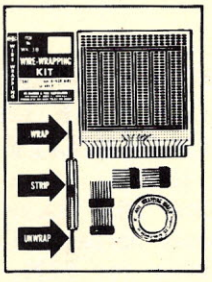
for quality electronic parts and tools.



WIRE-WRAPPING KITS

Contains: Hobby Wrap Tool WSU-30, (50 ft.) Roll of wire Prestripped wire 1" to 4" lengths (50 wires per package) stripped 1" both ends.

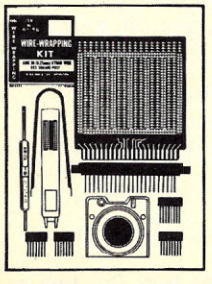
Wire Wrapping Kit, (Blue)	WK-2-B	$12.95
Wire Wrapping Kit, (Yellow)	WK-2-Y	$12.95
Wire Wrapping Kit, (White)	WK-2-W	$12.95
Wire Wrapping Kit (Red)	WK-2-R	$12.95



WIRE-WRAPPING KIT

Contains: Hobby Wrap Tool WSU-30, Roll of wire R-30B-0050, (2) 14 DIP's, (2) 16 DIP's and Hobby Board H-PCB-1.

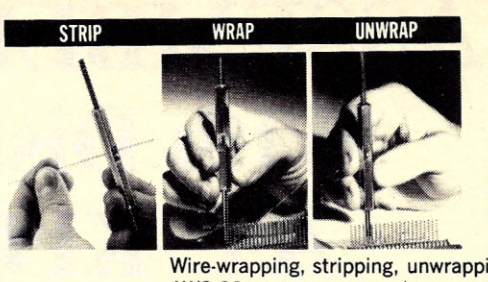
Wire-Wrapping Kit	WK-3B (Blue)	$16.95



WIRE-WRAPPING KIT

Contains: Hobby Wrap Tool WSU-30 M, Wire Dispenser WD-30-B, (2) 14 DIP's, (2) 16 DIP's, Hobby Board H-PCB-1, DIP/IC Insertion Tool INS-1416 and DIP/IC Extractor Tool EX-1

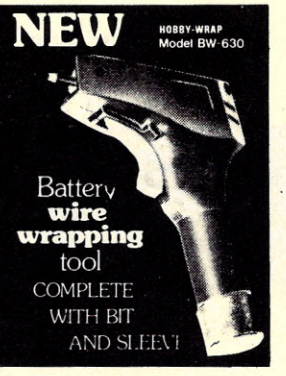
Wire-Wrapping Kit	WK-4B (Blue)	$25.99



HOBBY WRAP TOOL

Wire-wrapping, stripping, unwrapping tool for AWG 30 on.025 (0,63mm) Square Post.

Regular Wrap	WSU-30	$6.95
Modified Wrap	WSU-30M	$7.95



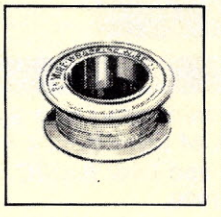
WIRE-WRAPPING TOOL

For .025" (0,63mm) sq. post "MODIFIED" wrap, positive indexing, anti-overwrapping device.

For AWG 30	BW-630	$34.95*
For AWG 26-28	BW-2628	$39.95*

Bit for AWG 30	BT-30	$3.95
Bit for AWG 26-28	BT-2628	$7.95

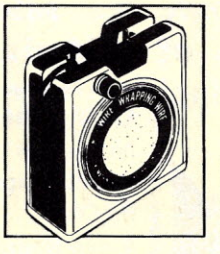
*USE "C" SIZE NI-CAD BATTERIES (NOT INCLUDED)



ROLLS OF WIRE

Wire for wire-wrapping AWG-30 (0.25mm) KYNAR® wire, 50 ft. roll, silver plated, solid conductor, easy stripping.

30-AWG Blue Wire, 50ft. Roll	R-30B-0050	$1.98
30-AWG Yellow Wire 50ft. Roll	R-30Y-0050	$1.98
30-AWG White Wire, 50ft. Roll	R-30W-0050	$1.98
30-AWG Red Wire, 50ft. Roll	R-30R-0050	$1.98



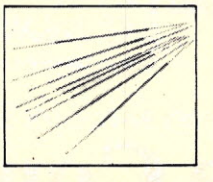
WIRE DISPENSER

- With 50 ft. Roll of AWG 30 KYNAR® wire-wrapping wire.
- Cuts the wire to length.
- Strips 1" of insulation.
- Refillable (For refills, see above)

Blue Wire	WD-30-B	$3.95
Yellow Wire	WD-30-Y	$3.95
White Wire	WD-30-W	$3.95
Red Wire	WD-30-R	$3.95

PRE CUT PRE STRIPPED WIRE

Wire for wire-wrapping, AWG-30 (0.25mm) KYNAR® wire, 50 wires per package stripped 1" both ends.



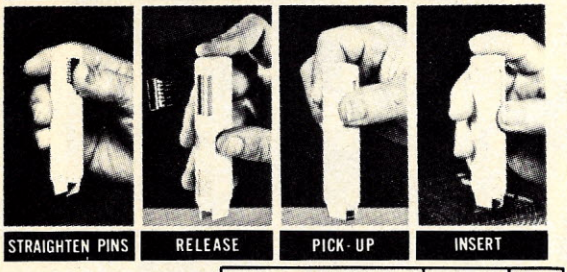
30-AWG blue Wire, 1" Long	30-B-50-010	$.99
30-AWG Yellow Wire, 1" Long	30-Y-50-010	$.99
30-AWG White Wire, 1" Long	30-W-50-010	$.99
30-AWG Red Wire, 1" Long	30-R-50-010	$.99
30-AWG Blue Wire, 2" Long	30-B-50-020	$1.07
30-AWG Yellow Wire, 2" Long	30-Y-50-020	$1.07
30-AWG White Wire, 2" Long	30-W-50-020	$1.07
30-AWG Red Wire, 2" Long	30-R-50-020	$1.07
30-AWG Blue Wire, 3" Long	30-B-50-030	$1.16
30-AWG Yellow Wire, 3" Long	30-Y-50-030	$1.16
30-AWG White Wire, 3" Long	30-W-50-030	$1.16
30-AWG Red Wire, 3" Long	30-R-50-030	$1.16
30-AWG Blue Wire, 4" Long	30-B-50-040	$1.23
30-AWG Yellow Wire, 4" Long	30-Y-50-040	$1.23
30-AWG White Wire, 4" Long	30-W-50-040	$1.23
30-AWG Red Wire, 4" Long	30-R-50-040	$1.23
30-AWG Blue Wire, 5" Long	30-B-50-050	$1.30
30-AWG Yellow Wire, 5" Long	30-Y-50-050	$1.30
30-AWG White Wire, 5" Long	30-W-50-050	$1.30
30-AWG Red Wire, 5" Long	30-R-50-050	$1.30
30-AWG Blue Wire, 6" Long	30-B-50-060	$1.38
30-AWG Yellow Wire, 6" Long	30-Y-50-060	$1.38
30-AWG White Wire, 6" Long	30-W-50-060	$1.38
30-AWG Red Wire, 6" Long	30-R-50-060	$1.38

® KYNAR-PENNWALT

MINIMUM ORDER $25.00, SHIPPING CHARGE $1.00, N.Y. CITY AND STATE RESIDENTS ADD TAX

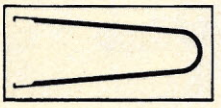
OK MACHINE & TOOL CORPORATION
3455 Conner St., Bronx, N.Y. 10475 ■ (212) 994-6600 ■ Telex 125091

DIP/IC INSERTION TOOL WITH PIN STRAIGHTENER



14-16 Pin Dip IC Inserter	INS-1416	$3.49

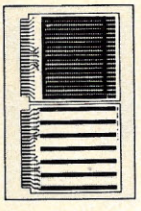
DIP/IC EXTRACTOR TOOL



The EX-1 Extractor is ideally suited for hobbyist or lab engineer. Featuring one piece spring steel construction. It will extract all LSI, MSI and SSI devices of from 8 to 24 pins.

Extractor Tool	EX-1	$1.49

P.C. BOARD



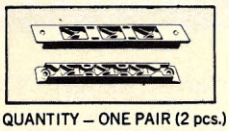
The 4 x 4.5 x 1/16 inch board is made of glass coated EPOXY Laminate and features solder coated 1 oz. copper pads. The board has provision for a 22/44 two sided edge connector, with contacts on standard .156 spacing. Edge contacts are non-dedicated for maximum flexibility.

The board contains a matrix of .040 in. diameter holes on .100 inch centers. The component side contains 76 two-hole pads that can accommodate any DIP size from 6-40 pins, as well as discrete components. Typical density is 18 of 14-Pin or 16-Pin DIP's. Components may be soldered directly to the board or intermediate sockets may be used for soldering or wire-wrapping.

Two independent bus systems are provided for voltage and ground on both sides of the board. In addition, the component side contains 14 individual busses running the full length of the board for complete wiring flexibility. These busses enable access from edge contacts to distant components. These busses can also serve to augment the voltage or ground busses, and may be cut to length for particular applications.

Hobby Board	H-PCB-1	$4.99

PC CARD GUIDES

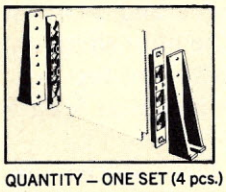


QUANTITY – ONE PAIR (2 pcs.)

TR-I consists of 2 guides precision molded with unique spring finger action that dampens shock and vibration, yet permits smooth insertion or extraction. Guides accommodate any card thickness from .040-.100 inches.

Card Guides	TR-1	$1.89

PC CARD GUIDES & BRACKETS

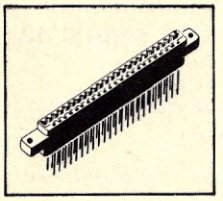


QUANTITY – ONE SET (4 pcs.)

TRS-2 kit includes 2 TR-1 guides plus 2 mounting brackets. Support brackets feature unique stabilizing post that permits secure mounting with only 1 screw.

Guides & Brackets	TRS-2	$3.79

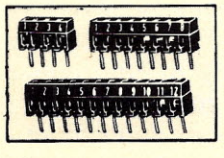
PC EDGE CONNECTOR



44 Pin, dual read out, .156" (3,96 mm) Contact Spacing, .025" (0,63 mm) square wire-wrapping pins.

P.C.. Edge Connector	CON-1	$3.49

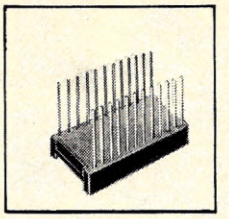
P.C.B. TERMINAL STRIPS



The TS strips provide positive screw activated clamping action, accommodate wire sizes 14-30 AWG (1, 8-0, 25mm). Pins are solder plated copper, .042 inch (1mm) diameter, on .200 incn (5mm) centers.

4-Pole	TS- 4	$1.39
8-Pole	TS- 8	$1.89
12-Pole	TS-12	$2.59

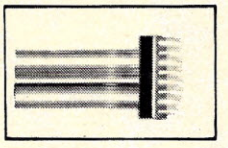
DIP SOCKET



Dual-in-line package, 3 level wire-wrapping, phosphor bronze contact, gold plated pins .025 (0,63mm) sq., .100 (2,54mm) center spacing.

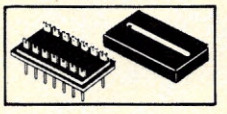
14 Pin Dip Socket	14 Dip	$0.79
16 Pin Dip Socket	16 Dip	$0.89

RIBBON CABLE ASSEMBLY SINGLE ENDED



With 14 Pin Dip Plug 24" Long (609mm)	SE14-24	$3.55
With 16 Pin Dip Plug 24" Long (609mm)	SE16-24	$3.75

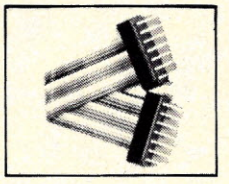
DIP PLUG WITH COVER FOR USE WITH RIBBON CABLE



14 Pin Plug & Cover	14-PLG	$1.45
16 Pin Plug & Cover	16-PLG	$1.59

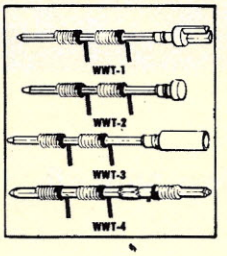
QUANTITY: 2 PLUGS, 2 COVERS

RIBBON CABLE ASSEMBLY DOUBLE ENDED



With 14 Pin Dip Plug -2" Long	DE 14-2	$3.75
With 14 Pin Dip Plug -4" Long	DE 14-4	$3.85
With 14 Pin Dip Plug -8" Long	DE 14-8	$3.95
With 16 Pin Dip Plug -2" Long	DE 16-2	$4.15
With 16 Pin Dip Plug -4" Long	DE 16-4	$4.25
With 16 Pin Dip Plug -8" Long	DE 16-8	$4.35

TERMINALS

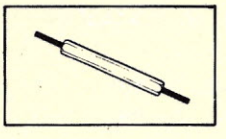


- .025 (0,63mm) Square Post
- 3 Level Wire-Wrapping
- Gold Plated

Slotted Terminal	WWT-1	$2.98
Single Sided Terminal	WWT-2	$2.98
IC Socket Terminal	WWT-3	$3.98
Double Sided Terminal	WWT-4	$1.98

25 PER PACKAGE

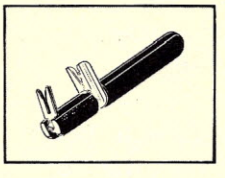
TERMINAL INSERTING TOOL



For inserting WWT-1, WWT-2, WWT-3, and WWT-4 Terminals into .040 (1,01mm) Dia. Holes.

INS-1	$2.49

WIRE CUT AND STRIP TOOL



Easy to operate . . . place wires (up to 4) in stripping slot with ends extending beyond cutter blades . . . press tool and pull . . . wire is cut and stripped to proper "wire-wrapping" length. The hardened steel cutting blades and sturdy construction of the tool insure long life.

Strip length easily adjustable for your applications.

DESCRIPTION	MODEL NUMBER	ADJUSTABLE "SHINER" LENGTH OF STRIPPED WIRE INCHES TO INCHES	Price
24 ga. Wire Cut and Strip Tool	ST-100-24	1¾" — 1¾"	$ 8.75
26 ga. Wire Cut and Strip Tool	ST-100-26	1¾" — 1¹¹⁄₁₆"	$ 8.75
26 ga. Wire Cut and Strip Tool	ST-100-26-875	⅞" — 1⅛"	$ 8.75
28 ga. Wire Cut and Strip Tool	ST-100-28	⅞" — 1⅛"	$11.50
30 ga. Wire Cut and Strip Tool	ST-100-30	⅞" — 1⅛"	$11.50

THE ABOVE LIST OF CUT AND STRIP TOOLS ARE NOT APPLICABLE FOR MYLENE OR TEFLON INSULATION.

MINIMUM ORDER $25.00, SHIPPING CHARGE $1.00, N.Y. CITY AND STATE RESIDENTS ADD TAX

OK MACHINE & TOOL CORPORATION
3455 Conner St., Bronx, N.Y. 10475 ■ (212) 994-6600 ■ Telex 125091

CIRCLE 117 ON READER SERVICE CARD

Fairytales

Allen Newell

Once upon a time, when it was still of some use to wish for what one wanted, . . .

> *. . . there lived a King and Queen who had a daughter who was lovely to behold, but who never laughed.*

Or perhaps:

> *. . . there lived an old fisherman by the side of a sea that had hardly any fishes in it.*

If you are like me, you are already hooked. You are ready to abandon all talk of present matters, of computers and electronic technology and professorships, and settle in to hear a fairy story. Their attraction reaches almost all of us.

They let us enter in upon an enchanted world. Magic abounds, though always in special ways. Animals talk, and not only animals but trees and bridges. Villainy is there, certainly danger. There are trials to be overcome — usually three of them. But there is always the happy ending. The spell is broken and the Princess smiles and marries the youth who made her laugh. The old fisherman gets the Jinni back in the bottle with the top on. And happiness is ever after, which means at least for a little while.

The experts tell us that fairy stories are for childhood. They contain lessons for the crises of growing up and their universal attraction comes because they deal with what is central to this universal time of life:

> *Like Hansel and Gretel, we have to leave home and find our own way.*
>
> *Like the Princess with the Frog King, we must learn to keep our word and embrace what we find ugly and disgusting, to discover that it contains our heart's desire.*
>
> *Or like Jack, in the story of the beanstalk, we can bring home the bacon if we persevere, even if our parents don't think we can.*

But there was more, if you remember your Jack:

Allen Newell, Computer Science Dept., Carnegie-Mellon University. Pittsburgh, PA. — Reprinted with permission.

First he escaped back home with a bag of gold. But Jack and his mother used up the gold, showing that one success is not enough.

Then he made a second trip up the beanstalk to the Giant's castle. This time he came home with the magic hen that lays golden eggs. Now Jack had a technology for satisfying his and his mother's wants.

But even so, material things are not sufficient for the full life. So on his third trip Jack brought home the golden singing harp, symbolizing the higher things of life.

The experts notwithstanding, fairy stories are for all of us. Indeed, this is true especially in our current times. For we are, all of us, children with respect to the future. We do not know what is coming. The future is to us as new, and as incomprehensible, as adult life is to children. We find ourselves troubled and fearful at the changes taking place in ourselves and our society. We need the hidden guidance of fairy stories to tell us of the trials we must overcome and assure us there will be a happy ending. Whether fairy stories have been written that speak to the heart of our own adult crises is not clear. How would we, the children, ever know? Perhaps we must get along with the fairy stories we have.

But even more, fairy stories seem to me to have a close connection to technology. The aim of technology, when properly applied, is to build a land of Faerie.

Well, that should come as a shock! The intellectual garb of the modern academic is cynicism. Like a follower in a great herd, as surely as I am an academic, I am a cynic. Yet I have just uttered a sentiment that is, if anything, straight from Pollyanna.

In point of fact, within the small circle of writers who manage to put technology and fairy stories between the same covers, the emphasis is always on the negative, on the dark side. The favorite stories are those that trouble:

Like the Sorcerer's Apprentice, who learns only enough magic to start the broom of technology hauling water from the River Rhine to the cistern, but who cannot stop it.

Like the Jinni in the bottle, where the story is never permitted to go to the conclusion in the Arabian Nights, with the Jinni snookered back into the bottle, but is always stopped with the Jinni hanging in air and the question along with it — Can we ever put the Jinni back? Or will there only be ink all over the sky 'til the stars go out?

Like the many stories of the three magic wishes, in which, promising infinite riches just for the asking, they are always spent, first on foolishness, second on disaster and third on bare recovery.

Recall the story of the Monkey's Paw, which came to the old English couple. Their first wish was for just 200 pounds. That was foolish. The second wish was for the return of their just killed son — whose accident had brought them a 200 pound award. That was disaster. The third wish was to send their son back to his opened grave, to try to recover for themselves a world where life could go on.

I see it differently. I see the computer as the enchanted technology. Better, it is the technology of enchantment. I mean that quite literally, so I had best explain.

There are two essential ingredients in computer technology. First, it is the technology of how to apply knowledge to action to achieve goals. It provides the capability for intelligent behavior. That is why we process data with computers — to get answers to solve our problems. That is what algorithms and programs are all about — frozen action to be thawed when needed.

The second ingredient is the miniaturization of the physical systems that have this ability for intelligent action. This is what Angel Jordan, my co-Whitaker Professor, has been telling us about in his talk. Computers are getting smaller, and cheaper, and faster, and more reliable, and less energy demanding. Everything is changing together in the right direction. The good things do not trade off against the bad ones. More speed does not mean more dollars. Small size does not mean lower reliability. On any given date, the expected painful tradeoffs do hold, just as we learned in elementary economics. It costs more to buy faster circuits or larger memories. But come back next year and everything is better: smaller, cheaper, faster, more reliable, less energy.

Thus computer technology differs from all other technologies precisely in providing the capability for an enchanted world:

For little boxes that make out your income tax for you.

For brakes that know how to stop on wet pavements.

For instruments that can converse with their users.

For bridges that watch out for the safety of those who cross them.

For streetlights that care about those who stand under them — who know the way, so no one need get lost.

In short, computer technology offers the possibility of incorporating intelligent behavior in all the nooks and crannies of our world. With it we could build an enchanted land.

All very good. But what about the Sorcerer's Apprentice? Two half-fallacies feed our fear that his nightmare might be ours. The first half-fallacy is that technologies are rigid and unthinking. Start the broom off carrying water and it does just that and not something else. But every computer scientist recognizes in the Sorcerer's Apprentice simply a program with a bug in it, embedded in a first generation operating system with no built-in panic button. Even with our computer systems today, poor things as they are, such blunderbus looping is no longer a specter.

Exactly what the computer provides is the ability to *not* be rigid and unthinking, but rather to behave conditionally. That is what it means to apply knowledge to action: it means to let the action taken reflect knowledge of the situation, to be sometimes this way, sometimes that, as appropriate. With small amounts of computer technology – that is, with small amounts of memory and small amounts of processing per decision – you often can't be conditional enough. That is certainly the story of the first decades of the computer revolution. It was too expensive and involved too much complexity to create systems with enough conditionality. We didn't know how and couldn't have afforded it if we had. Consequently, many applications were rigid and unthinking. It was indeed a Sorcerer's Apprentice who seemed to run the computerized billing service.

However, the import of miniaturization is that ultimately we will be able to have the capability for enough conditionality in a small enough space. And the import of our scientific study of computers is that we will know how to make all the conditionality work for us. Then the brooms of the world themselves can know enough to stop when things go wrong.

The second half-fallacy behind the Sorcerer's Apprentice is that technologies by their nature extract too high a price. That is a message of the recent literature of political ecology: Our technologies inevitably demand that we use up our precious world. There is rather abundant evidence for this view. Here in Western Pennsylvania, the price paid in enchantment of our countryside for taking our coal by strip mining is only too evident. Less in our awareness, because it was so thorough, was what the loggers did to Western Pennsylvania. Not once, but thrice, within forty years they swept the hillsides almost bare. The hot scalding breath of a dragon could hardly have done better for desolation.

But all is not inevitable. Ecologically, computer technology itself is nearly magic. The better it gets, the less of our environment it consumes. It is clean, unobtrusive, consumes little energy and little material. And as we push it to higher peaks of speed and memory, it becomes more of all these things. For deep technical reasons this has to be. There is no way to obtain immense amounts of processing power by freezing technology at some cost in dollars, material and energy per unit of computation, and then just buying more and more of it, consuming our wealth and our environment. Instead, for a long time to come, as we get more and more of it, the less will it impact our environment.

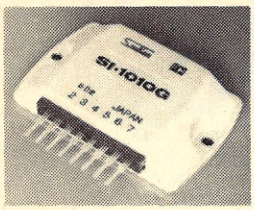
Even more, the computer is exactly the technology to permit us to cope intelligently with the use of our other resources. Again, by providing us with distributed intelligence, it can let us keep track of the use and abuse of our environment. And not only of the destruction that we ourselves visit on our world, but also that which nature does as well. Mt. Vesuvius was hardly bound by any antipollution ordinances posted on the walls of ancient Pompeii.

In sum, technology can be controlled, especially if it is saturated with intelligence to watch over how it goes, to keep accounts, to prevent errors, and to provide wisdom to each decision. And these guardians of our world, these magic informational dwarfs, need not extract too high a price.

But I said that the Sorcerer's Apprentice was guided by *half-fallacies*. I did not dismiss the view totally. Because, of course, in fairy stories there are great trials to be performed before the happy ending. Great dangers must be encountered and overcome. Because also, in fairy stories, the hero (or the heroine) – the one who achieves finally the happy ending – must grow in virtue and in mature understanding. No villians need apply for the central role. The fairy story that I am indirectly spinning here will not come true automatically. We must earn it.

Where are we now? We are not at the end of the story, though we are surely at the end of my talk. In fact the fairy story is hardly past its "Once upon a time". Still, I wish to assert that computer science and technology are the stuff out of which the future fairy land can be built. My faith is that the trials can be endured successfully, even by us children who fear that we are not so wise as we need to be. I might remind you, by the way, that the hero never has to make it all on his own. Prometheus is not the central character of any fairy story, but of a tragic myth. In fairy stories, magic friends sustain our hero and help him overcome the giants and the witches that beset him.

Finally, I wish to express my feeling of childlike wonder that my time to be awake on this earth has placed me in the middle of this particular fairy story. ■



SANKEN Series SI-1000G amplifiers are self-contained power hybrid amplifiers designed for Hi-Fi, stereo, musical instruments, public address systems and other audio applications. The amplifiers have quasi-complementary class B output. The circuit employs flip-chip transistors with high reliability and passivated chip power transistors with excellent secondary breakdown strength. Built-in current limiting is provided for SI-1050G and all devices can be operated from a single or split power supply.

SI-1010G (10W output) . . . $ 6.90
SI-1020G (20W output) . . . $13.95
SI-1030G (30W output) . . . $19.00
SI-1050G (50W output) . . . $27.80

Data Sheet with Application Notes — $0.50

SANKEN®

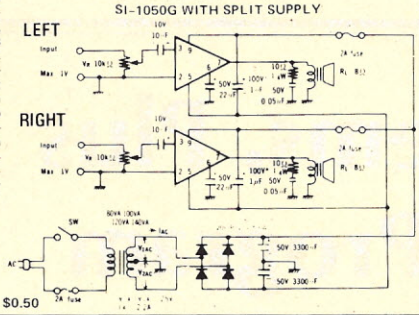
HYBRID AUDIO POWER AMPLIFIERS

- Multi-purpose linear amplifiers for comercial and industrial applicacations.
- Less than 0.5% harmonic distortion at full power level.
- 1/2 dB response from 20 to 100,000 Hz.
- Single or split (dual) power supply.
- Rugged, compact and lightweight packages.
- Built-in current limiting for SI-1050G and efficient heat radiating construction.

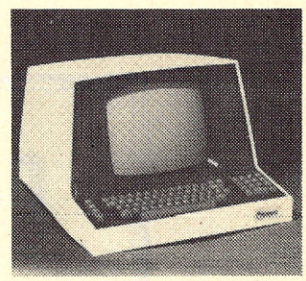
SOCKET FOR SANKEN AUDIO AMPLIFIERS

FOR 10 & 20 W. A-SI-8	$.95
FOR 30 & 50 W. A-SI-10	$.95

TYPICAL CONNECTIONS
SI-1050G WITH SPLIT SUPPLY



SOROC IQ 120 TERMINAL



FEATURES —

CURSOR CONTROL— Forespace, backspace, up, down, new line, return, home, tab, plus absolute cursor addressing.

TRANSMISSION MODES— Conversation (half and full Duplex) plus block mode—transmit a page at a time.

FIELD PROTECTION— Any part of the display can be "protected" to prevent overtyping. Protected fields are displayed at reduced intensity.

EDITING— Clear screen, typeover, absolute cursor addressing, erase to end of page, erase to end of line and erase to end of field.

DISPLAY FORMAT— 24 lines by 80 characters (1,920).

CHARACTER SET— 96 characters total. Upper and lower case ASCII.

KEYBOARD— 73 keys including numeric key pad.

REPEAT KEY— 15 cps repeat action.

DATA RATES— Thumbwheel selectable from 75 to 19,200 baud.

SCREEN— 12 inch rectangular CRT — P4 phosphor.

FULLY ASSEMBLED $995.00

50 VOLT CERAMIC DISC CAPACITORS
$1.00 Per Package

5pf	9/pkg	220pf	8/pkg	.001mfd	9/pkg	.015mfd	8/pkg
15pf	9/pkg	270pf	7/pkg	.0015mfd	9/pkg	.02mfd	8/pkg
25pf	9/pkg	300pf	7/pkg	.0022mfd	9/pkg	.022mfd	8/pkg
27pf	9/pkg	330pf	7/pkg	.003mfd	9/pkg	.03mfd	8/pkg
47pf	8/pkg	390pf	7/pkg	.0047mfd	9/pkg	.039mfd	7/pkg
68pf	8/pkg	470pf	7/pkg	.005mfd	9/pkg	.047mfd	7/pkg
100pf	8/pkg	560pf	7/pkg	.01mfd	9/pkg	.1mfd	6/pkg
150pf	8/pkg	680pf	7/pkg				

PLESSEY POLYESTER MINI-BOX CAPACITORS
$1.25 Per Package

MFD	QTY	MFD	QTY	MFD	QTY	MFD	QTY
.001	8/pkg	.0068	8/pkg	.039	7/pkg	.22	5/pkg
.0012	8/pkg	.0082	8/pkg	.047	7/pkg	.27	4/pkg
.0015	8/pkg	.01	8/pkg	.056	7/pkg	.33	4/pkg
.0018	8/pkg	.012	8/pkg	.068	7/pkg	.39	3/pkg
.0022	8/pkg	.015	7/pkg	.082	7/pkg	.47	3/pkg
.0027	8/pkg	.018	7/pkg	.1	7/pkg	.56	3/pkg
.0033	8/pkg	.022	7/pkg	.12	6/pkg	.68	2/pkg
.0039	8/pkg	.027	7/pkg	.15	6/pkg	.82	2/pkg
.0047	8/pkg	.033	7/pkg	.18	5/pkg	1.0	2/pkg
.0056	8/pkg						

CARBON FILM RESISTORS (5%)
Only in Multiples of 100 pcs per value (ohms)
¼W . . . $1.69 per 100
½W . . . $1.79 per 100

10	100	1.0K	10K	100K	1.0M
11	110	1.1K	11K	110K	1.1M
12	120	1.2K	12K	120K	1.2M
13	130	1.3K	13K	130K	1.3M
15	150	1.5K	15K	150K	1.5M
16	160	1.6K	16K	160K	1.6M
18	180	1.8K	18K	180K	1.8M
20	200	2.0K	20K	200K	2.0M
22	220	2.2K	22K	220K	2.2M
24	240	2.4K	24K	240K	2.4M
27	270	2.7K	27K	270K	2.7M
30	300	3.0K	30K	300K	3.0M
33	330	3.3K	33K	330K	3.3M
36	360	3.6K	36K	360K	3.6M
39	390	3.9K	39K	390K	3.9M
43	430	4.3K	43K	430K	4.3M
47	470	4.7K	47K	470K	4.7M
51	510	5.1K	51K	510K	5.1M
56	560	5.6K	56K	560K	5.6M
62	620	6.2K	62K	620K	6.2M
68	680	6.8K	68K	680K	6.8M
75	750	7.5K	75K	750K	7.5M
82	820	8.2K	82K	820K	8.2M
91	910	9.1K	91K	910K	10.0M

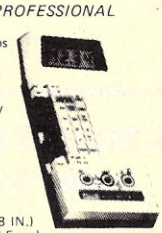
STANDARD MICROSYSTEMS
DATA COMMUNICATIONS CIRCUITS

COM2017H	UART (High Speed)	$ 9.00
COM2017HP	UART (High Speed)	5.70
COM2017P	UART	5.70
COM2502H	UART (High Speed)	14.20
COM2502HP	UART (High Speed)	9.00
COM2502P	UART	8.00
COM2601	USRT	30.00
COM5016	Dual Baud Rate Gen.	13.20
CRT5027	CRT Controller	50.00

MICROPROCESSOR CRYSTALS

FREQ (MHz)	CASE	P/N	PRICE	FREQ (MHz)	CASE	P/N	PRICE	FREQ (MHz)	CASE	P/N	PRICE
1.00	HC33	CY1A	$9.50	4.9562	HC18	CY6A	$6.00	15.00	HC18	CY15A	$4.75
1.8432	HC33	CD1A	9.50	5.00	HC18	CY7A	6.00	18.00	HC18	CY19A	4.75
2.00	HC33	CY2A	8.25	5.068	HC18	CY5B	6.00	18.432	HC18	CY19B	4.75
2.4576	HC33	CY2B	8.25	5.7143	HC18	CB5C	6.00	19.6608	HC18	CY20A	4.75
3.2768	HC33	CY3B	6.00	6.00	HC18	CY6B	6.00	20.00	HC18	CY22A	5.25
3.579	HC33	CY3D	6.00	6.144	HC18	CY6C	6.00	23.684	HC18	CY23B	5.25
4.00	HC18	CY3A	6.00	8.00	HC18	CY8G	6.00	27.00	HC18	CY27A	8.15
4.434	HC18	CY4C	6.00	10.00	HC18	CY12A	4.75	32.00	HC18	CY32A	8.15
4.55	HC18	CY4B	6.00	14.31818	HC18	CY14A	4.75	100.00	HC18	CY100A	8.15

NEW from FLUKE MODEL 8020A
THE DMM FOR THE PROFESSIONAL



- 200-Hr. Battery Life
- 26 Ranges for 7 Functions
- 2000 Count Resolution
- High Low Power Ohms
- Autozero and Auto polarity
- MOV protected to 6000V against hidden transients and overload protection to 300V AC
- Diode Test Function
- Conductance Function checks leakage resistance to 10,000 meg ohms
- Size HWL (7.1 X 3.4 X 1.8 IN.) (18.0 X 8.6 X 4.5 cm)
- Weight: 13 oz.

ONLY $169.00

★ E-Z-HOOK® ★

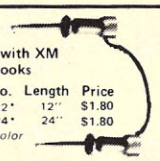
E-Z Hooks have been designed and field tested through the industry to save time and money in commercial electronic production and servicing. The spring-loaded hook attaches firmly, yet so gently it will not damage component . . . frees hands while testing. Durably constructed and fully insulated to a single contact point assuring true readings. Meets exacting laboratory and space age computer technology requirements. AVAILABLE IN 10 RETMA COLORS: Red, black, blue, green, oragne, yellow, white, violet, brown or gray.

MICRO HOOK

XM Micro Hook (1.75" long <1 gram) for difficult IC Testing. Permits hookups to delicate wires where weight and leverage may damage component . . . . . $.95 ea

Specify color. ORDER P/N XM

Jumper with XM Micro Hooks



Order No.	Length	Price
204XM-12*	12"	$1.80
204XM-24*	24"	$1.80

Specify color.

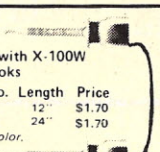
XM-S MICRO HOOK SET (Includes 1 ea. red, black, blue, green, orange, yellow, white, brown, violet and gray Micro Hook). At this low price you can afford more than one set.
COMPLETE SET (10) MICRO HOOKS $8.45

MINI HOOK

X100W Mini Hook (2.25" long) combines rugged construction, miniature size and Finger-eze Hypo Action for all the best test connections. Hook is large enough for component leads, yet small enough to get into tight places . . . . . $.80 ea

Specify color. ORDER P/N X100W

Jumper with X-100W Mini Hooks



Order No.	Length	Price
204-12W	12"	$1.70
204-24W	24"	$1.70

Specify color.

Jumper X-100W Mini Hook to Stacking Banana Plug

Order No.	Length	Price
201W	32"	$1.45

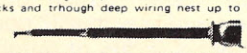
Specify color.

Jumper with X-100W Mini Hook

Order No.	Length	Price
201XM	32"	$1.50

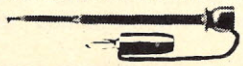
EXTRA LONG MINI HOOK

XL-1 Mini Hook (5.0" long) combines all the proven features of the X100W with an extra long body. It will make safe, short-free test connections in card racks and trhough deep wiring nest up to 4" . . ORDER P/N XL-1



Specify color. $1.30

JUMPER, XL-1 MINI HOOK TO STACKING BANANA PLUG

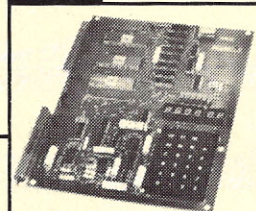


Order No.	Length
201XL-1	32"

Price $2.05

Specify color.

QUALITY PARTS

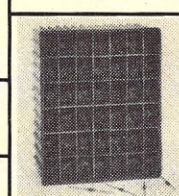


KIM-1 MICROCOMPUTOR

KIM-1—Computor Module from MOS Technology. 1K RAM, 2K ROM continuing system executive, complete audio cassett interface, 15 bidirectional I/O lines, a 24 key keyboard and a six-digit LED display.

Documentation—KIM-1 Users Manual 6500 Hardware Manual, and 6500 Programming Manual.

Fully Assembled Only
Fully Tested $245.00



HIGH QUALITY CARBON FILM RESISTOR KIT

COMPLETE WITH STORAGE BIN

Each KIT Contains 20 Each of 42 Different ¼ WATT 5% CARBON FILM RESISTORS from 68 ohm to 4.7 megohm

$24.90

ANCRONA

Send Check or Money Order to **P.O. Box 2208CC, Culver City, Calif. 90230.** Calif. residents add 6% sales tax. Minimum Order $10.00. ADD $1.00 to cover postage and handling. Master Charge and Bankamericard welcomed (include your card number and exp. date).

TELEPHONE ORDERS: Call (213) 641-4064

VISIT ONE OF OUR STORES TODAY
ANCRONA STORES DO NOT ACCEPT MAIL OR TELEPHONE ORDERS

CANADA, B.C. ANCRONA
5656 Fraser St.
Vancouver, B.C.
V5W 2V4
(604) 324-0707

CALIFORNIA ANCRONA
11080 Jefferson Blvd.
Culver City, CA 90230
(213) 390-3595

CALIFORNIA ANCRONA
1300 E. Edinger Ave.
Santa Ana, CA 92705
(714) 547-8424

ARIZONA ANCRONA
4518 E. Broadway
Tucson, AZ 85711
(602) 881-2348

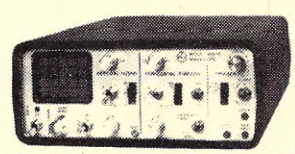
OREGON ANCRONA
1125 N.E. 82nd Ave.
Portland, OR 97220
(503) 254-5541

GEORGIA ANCRONA
3330 Piedmont Rd. NE
Atlanta, GA 30305
(404) 261-7100

TEXAS ANCRONA
2649 Richmond
Houston, TX 77098
(713) 529-3489

NON-LINEAR SYSTEMS, INC.

PORTABLE 15 MHz OSCILLOSCOPE



The MS-15 miniscope is only 2.7" x 6.4" x 7.5", and weighs only 3 lbs. Vertical bandwidth is 15 MHz. The graticuled rectangular viewing area is four divisions high by five divisions wide. Division spacing is 0.25 inches. Internal and external triggering, automatic and line synchronization modes, and a horizontal input are provided. There are 12 vertical gain settings from 0.01 V to 50 V per division, and twenty one time base settings from 0.1 μs to 0.5 μs per division. An optional 10 to 1 probe and a carrying case are also available. Power is provided by batteries or the 115 V, 60 Hz line.

MS-15	MINISCOPE	$289.00
41-140	CARRYING CASE	30.00
41-141	10 TO 1 PROBE	24.50

Intersil 3½ DIGIT PANEL METER

LCD or LED KITS

BUILD A WORKING DPM IN 1/2 HOUR WITH THESE COMPLETE EVALUATION KITS

Test these new parts for yourself with Intersil's low cost prototyping kits, complete with A/D converter and LCD display (for the 7106) or LED display (for the 7107). Kits provide all materials, including PC board, for a functioning panel meter.

ICL7106EV (LCD) $29.95 **ICL7107 (LED) $24.95**

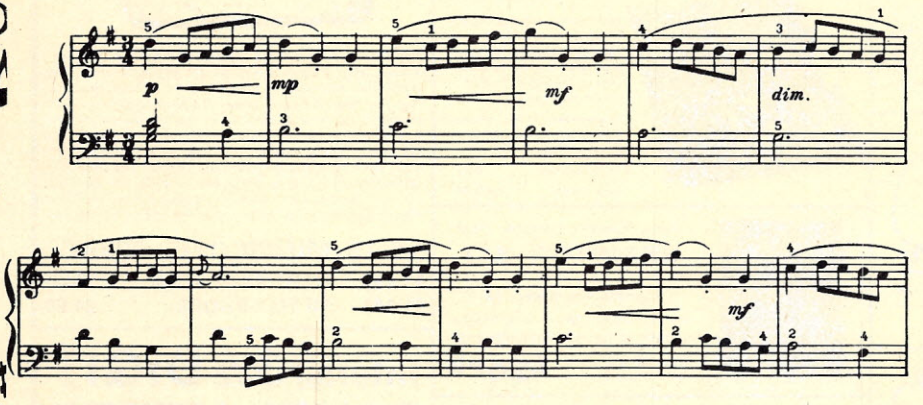
CIRCLE 150 ON READER SERVICE CARD

Computer Music

Four Systems, Plus a Music Program

Steve North

Bach's Minuet in G for Newtech Model 6
(using Steve North's program)




```
5 * MINUET IN G -- J.S. BACH
10 D2Q G1E A2E B2E C2E D2Q G1Q G1Q
20 E2Q C2E D2E E2E F#2E G2Q G1Q G1Q
30 C2Q D2E C2E B2E A2E
40 B2Q C2E B2E A2E G1E
50 F#1Q G1E A2E B2E G1E A2H.
60 D2Q G1E A2E B2E C2E D2Q G1Q G1Q
70 E2Q C2E D2E E2E F#2E G2Q G1Q G1Q
80 C2Q D2E C2E B2E A2E B2Q C2E B2E A2E G1E
90 A2Q B2E A2E G1E F#1E G1H.
100 B3Q G2E A3E B3E G2E A3Q D2E E2E F#2E D2E
110 G2Q E2E F#2E G2E D2E C#2Q B2E C#2E A2Q
120 A2E B2E C#2E D2E E2E F#2E
130 G2Q F#2Q E2Q F#2Q A2Q C#2Q D2H.
140 D2Q G1E F#1E G1Q E2Q G1E F#1E G1Q
150 D2Q C2Q B2Q A2E G1E F#1E G1E A2Q
160 D1E E1E F#1E G1E A2E B2E C2Q B2Q A2Q
170 B2E D2E G1Q F#1Q G1H.
```

Once upon a time (actually, a few years ago when personal computing was in its infancy) Steve Dompier had only an Altair 8800 with 256 bytes of memory, and no peripherals. Just for fun, he wrote a machine-language bubble-sort program. He noticed that when this program was run, strange patterns of noise could be heard on an AM radio he had placed on top of the computer. The radio, it seems, picked up the radio-frequency noise generated by the CPU. By executing the correct sequences of instructions, this noise could be made into musical patterns.[1] And so amateur computer music was born. Since then, the capabilities of personal-computer music systems have increased greatly. (However, the technique of playing CPU noise through an AM radio is still going strong. A recent issue of *Interface Age* carried a complete program used for the composition and execution of music using this method. The program included "Task Control Blocks," "Event Control Blocks," and "Supervisor Calls," which hopefully have those familiar with IBM operating systems rolling in fits of laughter. In addition, some of the Processor Technology video games generate

[1] Dompier, Steven. "Music of a Sort," *People's Computer Company*, Vol. 3, No. 5, pp 8,9.

Manufacturer	Solid State Music	ALF Products	Newtech	Software Technology
Model	SB-1	10-5-9	Model 6/68	Music System
Price per board	$199.95 kit	$159 kit $185 wired	$59.95 wired only	$24.50
Channels per board	1	4	1 with present software	3
Max. channels in system with current software	8	8	1	3
Stereo?	Yes	Yes, if you use two boards	No	No
Hardware Description	Digital synthesizer	Square-wave gen. with programmable frequency	DAC with audio amp	Bit-flipping picked up from interrupt signal on computer's bus
Software	MUS-X1 interpreter-excellent	Music playing program and subroutine—primitive to program	BASIC program and machine language subroutine—simple but usable	Music text editor and compiler software
Summary	High priced, but worth it—unmatched in sophistication	Interesting unit, but more software should be developed	Suitable for low-budget hobby applications	Inexpensive system with good software but limited performance in some areas—excellent purchase if you have a Processor Technology computer

nifty sound effects on an AM radio nearby. This is apparently intentional, since the TREK80 manual, for example, explains how to adjust the radio for the best sound).

One of the most inexpensive ways to generate computer music is to connect a speaker to a single bit on a parallel I/0 port. Turn the bit on (with an output instruction) and the speaker cone moves a little. Turn the bit off (by another output) and the cone moves back. Repeat this on/off cycle fast enough; that is, at audio frequencies, and the speaker is making audible sound. Of course, this is an extremely crude method for producing computer music. The computer is reduced to serving as a sort of programmable square-wave generator. And since the processor is responsible for every single wave produced, it can't spend much time doing other things. Further, as the complexity of the music software increases, the maximum frequency that can be generated decreases, because the overhead software takes up too much time between on/off-on/off cycles. If you just want to make funny sounds with your computer, this is great, but if you want real music, something more is required. (The Heath H-8 computer uses this idea to produce an audible beep when a command is entered, because it is cheap, and sophistication isn't needed).

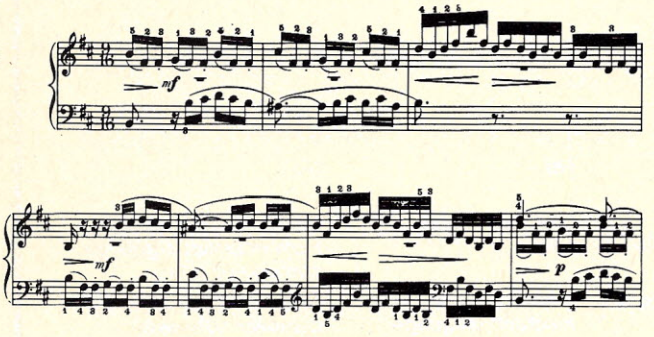
Since a number of music boards for microcomputers are now available commercially, we'll take a closer look at these. All of the boards are available for the S-100 bus. In addition, the Newtech Model 6 is available in a SWTPC 6800-compatible version, called the Model 68; and the ALF Products 10-5-9 board is available also in a form which connects to any parallel port, renamed the 10-5-10. In evaluating these music systems, we'll place special emphasis on the software involved. Although there are a number of bit hackers who are willing to build their own sophisticated music hardware from scratch and program it by converting sheet music to an obscure binary notation, most people would rather not approach computer-music transcription at such a low level—it seems to take all the fun out of making music with your own computer.

Solid State Music SB-1

The Solid State Music SB-1 is by far the most flexible and best performing board we've tried. Each board provides one channel of digital music synthesis, with complete control over frequency, volume, waveform, and envelope shape. The SB-1 is programmed through 256 bytes of on-card RAM which may be addressed anywhere in the highest 32K of memory (from 8000 hex to FF00 hex). However the Solid State Music software assumes that the cards are addressed beginning at 8000 hex. By writing into certain locations of this memory, various aspects of the SB-1's operation are controlled. Most of the memory is used for programming the main and envelope waveforms, by placing values corresponding to the desired amplitudes into sequential bytes of RAM. There are also four "control bytes" used to control the other functions. The byte at F0 sets the frequency produced within the current octave. Half of F1 controls the volume, while the other half controls the octave. F2 turns the board on or off, controls the envelope time, envelope function (continuous or one-shot), etc. One bit of F3 initializes the board.

A special wired header controls the number of waveforms within an envelope. The boards come configured for two waveforms per envelope and waveform select, which seems like a reasonable arrangement. This permits you to have two waveforms within the SB-1's memory, and to select one of the two for playing. Up to four waveforms per envelope may be used. According to the manual, "This allows a waveform

Bach's Invention in B Minor for Solid State Music MUS-X1
(using Solid State's program)



```
0010 "INVENTION IN B MINOR BY J.S. BACH"
0020 "TRANSCRIBED TO SOLID STATE MUSIC MUS-X1"
0030 "FINAL VERSION 9 JANUARY 1977 BY S.N."
0040 (150,9,16) (K,+C,+F)
0050 MIS4BFFGFFBFF; MI0.2BSR3B4CDC3B; MI0.RRR; MI0.RRR; MI0.RRR/
0060 S5C4FFGFF5C4FF; 0.3+AS+AB4C3B4C3+A; 0.RRR; 0.RRR; 0.RRR/
0070 T5D4B5DFBFDFD4B5D4BFBFDFD; 0.3BRR; 0.RRR; 0.RRR; 0.RRR/
0080 S3BRRR4B5CDC4B; S3BFFGFFBFF; 0.RRR; 0.RRR; 0.RRR/
0090 0.4+AS+AB5C4B5C4+A; S4C3FFGFF4C3FF; 0.RRR; 0.RRR; 0.RRR/
0100 T4BFB5DFD4B5D4BFBFDFD3B4D3B; T4D3B4DFBFDFD3B4D3BFBFDFD;
0110  0.RRR; 0.RRR; 0.RRR/
0120 P S4BFFGFFBFF; P 0.2BSR3B4CDC3B; P 0.5DO.D; P 0.RRR; P 0.RRR/
0130 0.4+GO.+G; 0.4ESR3EF+GFE; S5D4BB5C4BB5E4BB; 0.RRR; 0.RRR/
0140 S4AEEFEEAEE; 0.3ASR2AB3C2BA; 0.5CO.; 0.RRR; 0.RRR/
0150 0.4FO.; 0.3DSR2DEFED; S5C4AABAA5D4AA; 0.RRR; 0.RRR/
0160 S4GRRO.RR; 0.2GT3GBGEGECEC2A3C2A; T4B5D4BGBGO.ET5EGECEC;
0170  0.RRR; 0.RRR/
0180 (W1-1);(W1-2);(W1-3);(W1-1);(W1-1)/
0190 0.RRR; 0.2FT3FAFDFD2B3D2BGBG; T4A5C4AFAFO.DT5DFD4B5D4B;
0200  0.RRR; 0.RRR/
0210 (W1-0);(W1-0);(W1-0);(W1-0);(W1-0)/
0220 0.RRR; 0.2ET3EGECEC2A3C2AFAF; T4GBGEGECEC3A4C3AS4AFG;
0230  0.RRR; 0.RRR/
0240 S4D3AABAA4D3AA; 0.2DSRDEFED; 0.RRR; 0.RRR; 0.RRR/
0250 S4E3AABAA4E3AA; 0.2CSRAB3C2BA; 0.RRR; 0.RRR; 0.RRR/
0260 MI T4FDFA5D4B+GB+GE+GE3B4E3B+GB+G; MI 0.3DSDEFEFD;
0270  0.RRR; 0.RRR/
0280 P S4AEEFEEAEE; PO.3ESRAB4C3BA; 0.RRR; 0.3CRR; 0.RRR/
0290 S4BEEFEEBEE; 0.3GSGABABG; 0.RRR; 0.3ERR; 0.RRR/
0300 T5C4A5CEAE+DF+D4B5+D4BFBF+DF+D; 0.3ASAB4=C3B4=C3A;
0310  0.RRR; 0.2AQ.3F; 0.RRR/
0320 S4E3BB4=C3BB4E3BB; 0.3GSREFGFE; 0.4ERG; 0.RRR; 0.RRR/
0330 Q.3CO.; 0.3ASR2AB3C2BA; S4GEEFEEAEE; 0.RRR; 0.RRR/
0340 MP S4C3AABAA4D3AA; MP 0.3DSRDEFED; MP 0.4FO.; 0.RRR; 0.RRR/
0350 S4FDDEDDGDD; 0.3GSRGFEFG; 0.RRR; Q.3BO.; 0.RRR/
0360 Q.4EO.; 0.3CSRC2B+AB3C; 0.RRR; S3B+G+GA+G+G4C3+G+G; 0.RRR/
0370 I S4ECCDCCFCC; I 0.2FSR3FEDEF; I 0.RRR; I Q.3+AO.; I 0.RRR/
0380 II T4D3B4DFDFBFB5D4B5DO.G; II S3BRRO.RR; II 0.RRR;
0390  II S2BRRT3D2B3DFDFBGB4E3B4E; 0.RRR/
0400 T4C3A4CECEAEA5C4A5CO.F; 0.4AT3C2A3CECEAFA4D3A4D;
0410  0.RRR; 0.RRR; 0.RRR/
0420 T3BGB4D3B4DGDGBGBO.5E; T4GBGDGD3B4D3BGBGEGECEC; 0.RRR;
0430  0.RRR; 0.RRR/
0440 MI S5E4FFGFF5E4FF; MI 0.2+AS+AB3C2FBA; MI 0.RRR;
0450  MI 0.RRR; MI 0.RRR/
0460 Q.5DO.D; 0.2BRB; S4F3BB4D3BB4F3BB; 0.RRR; 0.RRR/
0470 S5D4BB5=C4BB5G4BB; Q.3EO.E; Q.4GO.G; 0.RRR; 0.RRR/
0480 0.4+AQ.5F; 0.3EQ.3E; 0.4FQ.A; 0.RRR; 0.RRR/
0490 T5F4B5DFBFDFD4B5D4BFBFDFD; O3DSRO.RR; 0.RRR;
0500  0.RRR; 0.RRR/
0510 Q.5DO.5D; T3BGB4DGD3B4D3BGBGDGD2B3D2B; Q.4BO.4B; 0.RRR; 0.RRR/
0520 S5DCC+GCCDCC; 0.2+ERR; Q.4BO.4B; 0.RRR; 0.RRR/
0530 S5=ECC+ACDEDC; 0.2FSR3EFGFE; 0.3ASRAB5C4BA; 0.RRR; 0.RRR/
0540 S5FGEO.DXCDCDCDS4B; O3DSEOFSEOFSF; Q.4BO.+A; 0.RRR; 0.RRR/
0550 H.4B; H.2B; H.4B; 0.RRR; 0.RRR/
0560 /L
```

The above is a transcription of "Invention in B Minor" (by J.S. Bach) to the Solid State Music language, SB-1. A portion of the original sheet music is also presented here, and a description of what particular portions of the program mean.

Lines 10-30: These are comments, enclosed in quotes. The interpreter ignores. them.

Line 40: This line initializes the time signature (150 beats per minute, 9/16 time) and key signature (B Minor, C and F sharp).

Lines 50-550: These are the actual music lines. The measures are separated by slash signs (/) and the voices within the measures by a semicolon (;). In line 50, the first characters of the first voice, MI, indicate that the voice is to be played with moderate intensity. (Further down, in line 120, it is changed to soft with the letter P). The next character in the line is S, which means that sixteenth notes are to be played. Then comes a 4, which means that the following notes are in the fourth octave, and following is a series of notes, first B, then F, and so on. The last three voices of this first measure aren't used, so after setting them to moderate intensity, rests are played. Lines 180 and 210 were inserted rather hastily, to show how to use the waveform control commands. In line 180, waveform number 2 is sent to the first memory half of voice number two (the card at 8100 hex), which is the one being used to play music. The waveforms were changed rather capriciously in line 180 and are changed back to normal in line 210.

Line 560: The /L indicates the end of the piece.

---

to be predistorted with added harmonics at the beginning of an envelope and then to be almost pure sinewave at the end of the envelope."

Fortunately it is never necessary to program the SB-1 at a low level. Solid State Music provides an excellent music interpreter with the SB-1, called MUS-X1. MUS-X1 permits direct transcription of sheet music and high-level control of the SB-1's special functions. The table shows the symbols used.

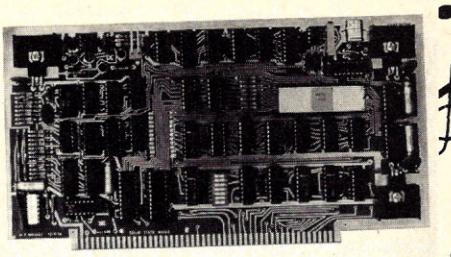
Since each SB-1 card produces only one voice, several cards are needed to create sophisticated music. (Our system has five cards, which are enough for almost all music). The interpreter supports up to eight cards. Like the SB-1 hardware, MUS-X1 is quite versatile, but easy to use. MUS-X1 comes with a built-in table of waveforms and envelopes, so there is no need to write your own unless you want to. You might ask why the interpreter needs to have a table of waveforms. This permits you to program several cards using a small "library" of up to eight waveforms, quickly and easily. After setting initial parameters in the music (such as key signature and time signature) you merely start coding the music as you read it, voice by voice. If you wanted to play four middle C's as quarter notes, you'd type Q4CCCC which means: play quarter notes, fourth octave, and four C's. You need only code most parameters when you want them to change. In other words, you don't have to tell the interpreter that all the notes are quarter notes four times—just once. This is a great improvement over other systems, which would have forced you to specify a duration and octave for all the notes.

MUS-X1 seems to have been very well thought out. Since the processor

has a very high level of independence from the music-producing process (it need only intervene when the SB-1 has to be reprogrammed for a new note, and not for every milisecond of actual sound), the interpreter isn't computing that much, but spends a lot of time in a software delay loop, while the hardware does its thing. The software contains several entry points: one for playing the music with a *software* timer as described, and others for playing music with a *hardware* timer which would interrupt the CPU when necessary to reprogram the SB-1s, but which would also permit the processor to do something significant when it wasn't needed to control the music. This would let you play Star Trek or edit another music file while listening to your music system, and also increase the stability of the interpreter's timing (which is already good). All the software needed to do this is already present in MUS-X1, so no doubt Solid State Music will offer a suitable interrupt card in the future.

The documentation for the SB-1 and MUS-X1 is also quite good. The documentation of MUS-X1 includes a source listing of the interpreter and examples of use of the language. However, the documentation does not explain how to write your own waveforms, which seems to be a tricky thing. Perhaps practice (and a little more know-how) helps. The object code for MUS-X1 is assembled at 4000 hex (starting at 20K). The reason for this was to allow other user software to reside lower than MUS-X1, especially handy if you're using MUS-X1 in the interrupt-driven mode. The interpreter has an entry point that causes it to assume that the music file is located at 5000 hex (24K and up) but there are other entry points that permit location of the file elsewhere. Solid State Music also provided some excellent demo music for the SB-1.

There are two things about the SB-1 of which potential buyers buyers should be aware. First, it is on the expensive side, compared with what most amateur computer hackers are used to paying for components. Since each card only handles one voice, several are necessary. However, for someone seriously interested in computer music, the price is quite attractive compared with $25,000 systems. Second, the music interpreter is an interpreter only; it expects the music-source file to be loaded in memory,



Solid State Music SB-1 synthesizer board, one channel per board, with full control of frequency, volume, waveform, and envelope shape.

waiting to be played. So an external text editor is required. This permits you to use whatever text editor you may already have to create MUS-X1 source programs. According to the manual, MUS-X1 expects the music file to have lines in the format: line number, text, carriage return. However we found that files written in the Processor Tech file format (which is basically the same as the above, but also includes the binary number of characters in the line in front of the line number) to be compatible with MUS-X1.

MUS-X1 has a nifty way of informing you of errors in the music file. Rather than print the error, it plays the portion with the error all by itself, so that by listening to the music and following it as written normally, you can easily determine the errors. For instance, if one voice in a measure has too many notes, that voice would be played all by itself, then a short pause, then the whole measure. This is much preferable to ordinary printed messages.

How does it sound? That's a very hard question to answer in writing. The SB-1 does definitely sound like a synthesizer and not like a sick sinewave generator. On the other hand, it may not be up to the level of everything in Switched-On Bach, since after all it is a digital synthesizer connected to an 8-bit micro. Nevertheless of all the units we've tried, it is unsurpassed in ability, and should be able to keep anyone with more than a passing interest in computer music happy for quite a while.

ALF Products 10-5-9

The 10-5-9 is a recent and comparatively unknown entry in the amateur computer-music field. (At least, I didn't know it existed until our publisher, Dave Ahl, handed me one to try out. However, I did know that ALF markets software in HP 2000 BASIC, such as a two-terminal Star Trek game. The 10-5-9 is a "quad chromatic pitch generator" which means that it is capable of generating a full 12-note scale (both white *and* black notes on the piano)

Solid State Music MUS-X1 Symbols

Symbol	*Meaning*
Letter A-G	Letter name of a note to be played
Number 0-9	Select octave number
+, -, =	Sharp, flat, natural (immediately preceding a note, as in +C, -D, or =E)
W,H,Q,O,S,T,X	Duration of note, from whole note through sixty-fourth note.
.	Increase duration by one-half
R	Rest
(X,Y,Z)	Time signature: X beats per minute, Y beats per measure, Z note gets one beat
(K,arguments)	Key signature
PP, P, MP,MI,I,II,Z	Loudness command, from *very soft* to ZAP!
Rx	Repeat x times
RO	Marks end of repeat section
J	Marks exit point in repeat section to use as the new ending
L1,L0	Legato and cancel-legato commands
L	End of music indicator ("Last")
(Fx)	Play frequency x in the current octave
(Tx)	Set envelope duration to x
(Vx)	Set volume to x
(WO x,y. . .)	Loads the interpreter's internal waveform table number x with the waveform specified by y. . .
(Wx-y)	Loads waveform number y into memory-half x of the SB-1 card (where x is 1 or 2)
(Wx)	Select memory-half x for playing music (where x is 1 or 2)
(EO-x,y. . .) (Ex-y) (Ex)	Essentially the same as WO and Wx waveform-control commands, but control envelope shape
;	Separates voices within a measure
/	Separates measures

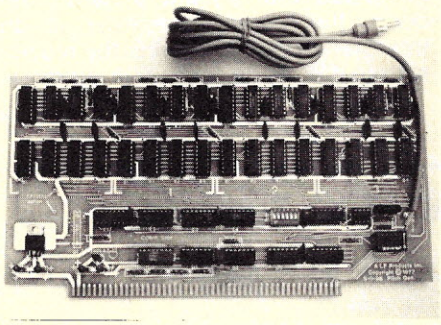
and has four voices on one board. No provision is made for control of the envelope or main waveform shapes, though there is a header that interfaces the 10-5-9 to other audio devices for altering the sound produced.

The 10-5-9 hardware is able to operate with a great deal of independence from the processor, which is desirable. Each channel is controlled by a separate I/0 port (with the first channel at 80 hex, the second at 81, and so on). To make a channel play a note, the corresponding value is output to that port. For instance, to play a middle C on channel #0 (the first one) you would output a 303 octal to port 80 hex. The channel will continue to play this note until told to do something else. Although the board doesn't allow control of the main waveform or envelope (and thus is not nearly as sophisticated as the SB-1) its sound is quite pleasant to listen to.

The manual contains a good description of how to build the board and how it works. There's an interesting note near the start of the assembly instructions, which says, in effect, "Wait! You can still send this back and get one assembled!" But assembly of the 10-5-9 (should you buy a kit) appears to be easy, since the board layout is quite straightforward. There is a section in the manual on reading sheet music, but nothing about taking sheet music and actually playing it on your 10-5-9. There is no music language provided with the board (yet) so the manual contains a list of the binary equivalents of chromatic pitches. Obviously, to play music you need to feed the board the right values but they must also be output in the proper sequence and changed at the proper rate.

But all is not lost! ALF did send a copy of some music-playing software with the 10-5-9. This software consists of two programs: a main program, and a subroutine, which load at 0000 and use 1K of memory. The main program reads in a music file on papertape, displays the title of the piece on a VDM-1 screen, and also draws three rows of "-" signs on the screen. The title of the song and other information is contained in a header block on the papertape. The main program then calls the subroutine 256 times per second, to play the music, and flashes black squares on the "keyboard" of minus signs, to indicate the notes being played (sheer gimmickry, but very entertaining and ingenious nevertheless). The format of the music as represented on the papertape is not high-level. For example, values 0 through 95 are used for tones, with 0 for "A0" (the lowest A) up to 95 for "GS7" (G-sharp in the seventh octave). Other values indicate a rest, set the "transpose register," etc. We did not even attempt to code our own music. However ALF kindly sent us about a dozen fairly sophisticated compositions to demonstrate the capabilities of the 10-5-9. As it turns out, ALF is presently developing a compiler program that will convert high-level source in a music language into this low-level papertape format. Although ALF is already using this compiler in-house, they said (at the time of this writing) that it isn't ready for release yet. Well, it's encouraging that they're not trying to rush an untested product into production. (I should mention at this point that although this lack of a high-level music language is important, ALF does document what software it has to the hilt).

The ALF music software is designed to be used with a timer which interrupts the CPU 256 times per second. When the CPU is interrupted, control is transferred from the main program to the subroutine to play the music. Obviously, having a hardware timer improves the stability of the music timing, but I didn't want to buy an interrupt timer or kluge one of my own.



ALF Products 10-5-9 Quad Chromatic Pitch Generator, with four voices on one board and a full 12-note scale.

I found that it is rather simple to add a software timing loop to the main program, which creates a time delay and then calls the subroutine, rather than letting the interrupt-timer hardware handle this. The time stability seemed satisfactory, though I have no idea how close my software timing loop was to being 1/256th of a second long. A quantitative test isn't very difficult, but I merely adjusted it by ear. However, to do this I loaded the source code for the program and re-assembled it with my own fixes.

Incidentally, ALF has an "AD8 Micro-Bus Synthesis Board" which is a full-scale synthesis system controlled by a computer; it appears from the documentation that this also generates chromatic tones only (no quarter-tones or such), and that the main waveform control is purely digital, while the envelope shape (rise, fall, and sustain) is analog. I'm afraid to ask what an AD8 Micro-Bus is, but the unit can be controlled by an adapter board in any computer. No mention of any software, though.

Since the 10-5-9 provides four channels on one board, it should be of interest to the hobbyist who is on a limited budget and who is willing to settle for less performance. If you're a real hardware hacker, you may wish to use the 10-5-9 as the basis for your own custom-designed music system. We predict that the 10-5-9 will become a significant product (since many people aren't serious enough about computer music to get the SB-1) when ALF releases its music compiler.

Newtech Model 6

The Newtech Model 6 is the simplest music board we tested. The hardware consists of a latched parallel port that controls a DAC (digital-to-analog converter) which in turn drives an op amp. The op amp is connected to a small on-board speaker, or to an external amplifier. To produce tones, one merely outputs varying values to the control port, which causes the speaker cone to change position (because of the change in the voltage output by the DAC). This is a simple way of generating music, similar to the bit-flipping method described earlier. In this case, however, you tell the speaker cone how far to move. Therefore, in playing music, the processor must spend significant amounts of time telling the cone to move from one position to another, and can't spend too much time doing other things. In fact in actual practice the amplitude control is used for the envelope shape, and the main waveform is a square wave.

The Newtech music playing software is composed of two programs: Microscore, and Microplay. Microscore is a BASIC program which takes a source file written in a music encoding system, which produces object in the form of a table of notes to be played, in binary. To make the music source file accessible to BASIC, the code representing each note is placed in a BASIC data string. Microscore processes these strings and places the binary object at an arbitrarily-chosen absolute memory location. After this binary table is made, Microplay makes the music itself. Because of the timing limitations involved, Microplay is a machine-language program. (Incidentally, I tried a BASIC program that made square waves at the highest possible frequency, and found, as one would expect, that the BASIC interpreter consumes too much time between half-waves for this arrangement to have a usable frequency range). Microscore allows only one envelope shape. The format for each note is as follows:

First character: A-G (letter name of the note)

Second character: # (sharp), - (flat) space (natural)

Third character: 1,2, or 3 (octave)

number)

Fourth character: W,H,Q,E,S (duration of note, from whole note to sixteenth note)

Fifth character: . (optional; increases duration by half)

So, for instance, to encode middle G as an eighth note, you'd write: G 2E

Newtech recently distributed a newsletter that contained improved versions of Microscore and Microplay. Revision B of this software allows up to eight envelopes, which are defined as tables in Microscore and thus not controllable from the BASIC program. One of the envelopes is a rest, which is a very useful thing to have! So if you're using a Model 6 and don't have the latest revisions of the software, try to get them. The format for the new music notation is very similar to the old, except that an extra optional character is added to the beginning of the note to indicate a new envelope choice.

In the opinion of the writer, it is rather annoying to have to enter all the notes as character-string data in BASIC, and to have to include an unnecessary space in the middle of many notes. In addition, Microplay must be loaded separately from the BASIC interpreter and program that process the music source file. Therefore I wrote a BASIC program which serves as a complete music compiling and playing system for the Model 6. This software permits the music data to be typed in with a variable number of notes on a line (separated by spaces) with a line-editing system similar to BASIC itself. The program prompts with a "?" (obviously a BASIC INPUT statement) and then accepts a program line (starting with a line number) or a command. The allowable commands are:

LIST x1 x2: Lists the music file with optional starting and ending line numbers.

NEW: Clears out the workspace.

SAVE "name": Saves the music file on cassette, with the specified name. This particular version uses the Tarbell cassette but the only requirement of this routine is that it save the array P$ out on a mass-storage device and later be able to yank it back in.

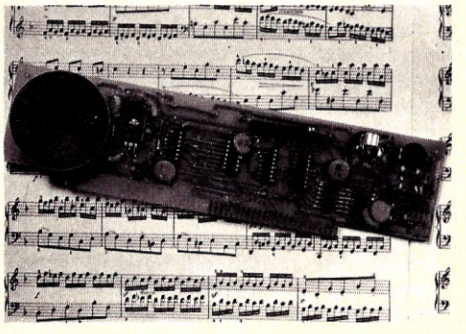
LOAD "name": Loads the named music file after clearing out the workspace.

PLAY: This command causes compilation of the music file and then playing of the music. The format of the music notes is the same as that used by the original Newtech software, so there is no envelope control, except that if a note is a natural, no space is needed in the middle of a note. The notes are compiled, and then a copy of Microscore (in BASIC data statements) is loaded into memory. Then this copy of Microscore is called from BASIC to play the music that was just compiled. A merger of this program and the revised version of Microscore/Microplay wouldn't be very tough.

This program runs in a 32K machine with MITS Extended BASIC. However, it should be possible to run this program in a 24K or possibly 20K machine. First of all, Extended BASIC uses 14.6K of memory, while 8K BASIC uses less than 8K of memory, and this program uses very few Extended BASIC features (only hexadecimal notation for constants and HEX$ to print hex values, I believe). Second, the space-consuming comments could be removed.

For other software, the manual recommends Malcolm Wright's "Alphanumeric Music with Amplitude Control." Though this is an excellent source of information, we noted with some irony that Malcolm Wright is the designer of both the Solid Sate Music SB-1 and MUS-X1.

Although there is an on-board speaker, it isn't much use when the small board is buried deep inside a computer among other boards, with a fan running and a noisy Teletype nearby. It is generally necessary to use an external speaker or amplifier (as we did) to get acceptable sound. The speaker is nice for test purposes, though.



Newtech Computer Systems Model 6 Music Board, available in both S-100 and Southwest 6800-compatible versions.

The Newtech Model 6 music board is not a high-priced, high-performance synthesizer, but it is a nice little music board that would be adequate for the needs of most hobbyists. Because of the technique of music production used, Microscore/Microplay generates only one voice. However, our Newtech source says that they are shortly releasing software that will generate two voices and permit user control of envelopes (we've already heard it in action). Newtech feels that one of the advantages of writing most of the software in BASIC is that it is highly transportable, so that the same software with a few modifications may be used on either an IMSAI 8080 or a SWTPC 6800, and only the machine-language Microplay program has to be rewritten. Additionally, Newtech is thinking seriously about the development of CAI music software for both home and educational use. The unit we had to try out did make slight clicking noises as the envelope shape was changed but, according to Newtech, this is an uncommon phenomenon reported on only a few boards. Newtech's demo system did indeed sound much better, and they were able to suggest a simple fix for this problem. So, although the Model 6 is not really a tremendously complex piece of hardware, it would do the job for most hobbyists.

Other Systems

Two music systems are not directly covered in this review. One is the Software Technology Music System, described in the Sept-Oct 1977 issue of *Creative.* This system, which costs $24.50, consists of a small S-100 bus board with a handful of resistors and capacitors on it, which is connected to an audio amp. Tones are generated by wiggling the voltage on the PINTE line of the S-100 bus. The software consists of a text editor and a music compiler for a high-level music language. The compiler apparently converts the source code into the proper instructions to manipulate the PINTE line. This music system seems to be designed primarily for use with the Processor Technology SOL computer, since the software makes calls to the SOLOS monitor, and the software is provided in a SOL-compatible format. This system could be used with another computer, but you would need a CUTER (Processor Tech) cassette interface, or a Kansas City/BYTE interface, and a monitor to provide the same functions as SOLOS. According to experts we've met at computer shows, it sounds like a primitive reed organ. All we can say is that after hours of exposure to this music, we find it rather unbearable, but this could be true of any music system.

The other system is a music board made by the company that started the whole amateur-computer phenomena rolling, MITS. The writer has not seen any information on this unit, but Dave Ahl reports that he saw and heard one at a computer conference in Toronto and that it is quite impressive. There will probably be more information on this product in *Creative* as it becomes available.

Please note that *all* the systems tested require an external amp. It is just impractical to try to provide amplification on a small printed-circuit board when most people have a high-quality stereo amplifier anyway. Keep this too in mind if you've spent all your hard-earned bucks on a computer and don't have any kind of audio amp to use with your music system.

Newtech Music Program and Description

Program Description

Lines 10-140: Initialization section. When you start BASIC, you must reserve some memory for storing the compiled music code and loading and running Microplay. Variable M9 is set to the address of the top of this memory. N is the maximum number of music program lines plus one. M is the number of lines actually in the music program, plus one. Arrays P and P$ hold the music program line numbers and the text, respectively. P(M) is always 10000; the highest legal line number is 9999.

Lines 150-230: Here's where we get an input line from the user. If the input starts with a command keyword (NEW, PLAY, LIST, etc.) then control is passed to the appropriate routine. If not, the program continues with the next section.

Lines 240-350: If we get this far, then it is assumed that the input was a music program line. Therefore, an attempt is made to get a line number from the input line. If none is found then an error message is printed. Otherwise, the program determines where this line is in relation to the line numbers of the music program.

Lines 360-470: This part of the program is used when the input line number matches an existing line in the music program. The operation must therefore be a delete (if there are no more characters in the input line) or a replacement (if there are more characters). In the event of a replacement, the text in the input line is merely inserted in place of the old text. If there is a deletion, all the lines and line numbers above the line to be deleted are shifted down one position, thus overlaying the deleted line.

Lines 480-590: This section is used when the input line number is between two existing lines. The lines and line numbers above the line to be inserted are shifted up one (to make room for the new line) and the line is inserted in the middle of the music program area.

Lines 600-670: Since one frequently enters lines in numeric sequence (10, 20, 30...) this routine improves response time. If the new line number is greater than the highest line number in the music program, this part of the program adds the new line to the end.

Lines 1000-1040: NEW command. The arrays are re-initialized by calling the subroutine at 8500.

Lines 2000-3190: PLAY command. The first part of this program isolates the individual notes in the string array. The notes are separated by spaces. This routine scans through the music program using X to hold the current line number and Y to hold the current character number on the line. Notes are placed in W$ for further processing. If a * is encountered, the rest of the program line is ignored.

The program then "compiles" the note in W$, using a technique very similar to that used in Microscore, except that a space is not needed if a note is a natural. The compiled code is poked in starting at the absolute memory location defined in line 2110. This location will have to be changed to conform with whatever area of memory you have set aside. When the compilation is complete, an end-of-score marker, binary zero, is poked in.

The next task is to load in a copy of the machine-language Microplay program after the binary music table. The code for Microplay is stored in BASIC data statements, which hold the bytes to be loaded. Since it is necessary to locate some 16-bit address constants, 1000 is added to the first byte of a two-byte address constant. The 1000 is subtracted and the next two values are added to the address where we began loading Microplay. Then a few pointers must be filled in. One of these is the pointer within BASIC which tells it where the machine-language subroutine (Microplay) is located. This is done in line 3060 with the DEFUSR command. Then it is necessary to inform the machine-language subroutine where the compiled music table is located in memory; this is done in line 3090-3110. The user subroutine (Microplay) is then called from BASIC to play the music.

Lines 4000-4250: This handles the LIST command. The command may have a starting and ending line number, just a starting line number, or no line number (in which case the whole file is listed). The line number is printed as a character string to improve the formatting.

Lines 5000-5470: The SAVE command. This routine is rather hardware-dependent. If you have a disk or good cassette I/0 from BASIC, by all means use it. (I happened to have a Tarbell cassette, nonstandard I/0, and no

Author's Program Using Newtech Hardware

```
10 REM
20 REM     MUSIC SYSTEM FOR NEWTECH MUSIC BOARDS
30 REM     STEVE NORTH
40 REM     CREATIVE COMPUTING
50 REM     PO BOX 789-M MORRISTOWN NEW JERSEY 07960
60 REM
70 REM     WHAT A WAY TO WRITE A COMPILER
80 REM
85 CLEAR(3000):REM SET STRING SPACE
90 PRINT "NORTH'S MUSIC SYSTEM VERSION 2.1"
100 PRINT
110 M9=32000: REM  SET M9 TO ADDR OF HIGHEST USABLE BYTE
120 N=100
130 DIM P(N),P$(N)
140 M=1:P(M)=10000
150 REM
160 REM      -----COMMAND INPUT ROUTINE-----
170 REM
180 INPUT C$
190 IF C$="NEW" THEN 1000
200 IF C$="PLAY" THEN 2000
210 IF LEFT$(C$,4)="LIST" THEN 4000
220 IF LEFT$(C$,4)="SAVE" THEN 5000
230 IF LEFT$(C$,4)="LOAD" THEN 6000
240 REM  IF WE GET TO HERE, C$ MUST BE A PROGRAM LINE
250 REM  SO WE TRY TO GET THE LINE NUMBER
260 L=1
270 GOSUB 9000
280 IF E=1 THEN PRINT "ERROR: MEANINGLESS INPUT":GOTO 150
290 IF P=-1 THEN PRINT "ERROR: ILLEGAL LINE NUMBER":GOTO 150
300 REM  NOW, TRY TO FIND LINE NP (I'M LKING THROUGH YOU)
310 IF M>1 THEN IF P>P(M-1) THEN 600
```

```
320 FOR X=1 TO M
330 IF P=P(X) THEN 360
340 IF P<P(X) THEN 480
350 NEXT X
360 REM      -----REPLACE OR DELETE EXISTING LINE-----
370 IF L=-1 THEN 400
380 REM  MUST BE REPLACEMENT
390 GOTO 650
400 REM  COME HERE IF DELETION OF EXISTING LINE
410 FOR Y=X+1 TO M
420 P(Y-1)=P(Y)
430 P$(Y-1)=P$(Y)
440 NEXT Y
450 P(M)=0
460 M=M-1
470 GOTO 150
480 REM      -----INSERT NEW PROGRAM LINE-----
490 IF L=-1 THEN PRINT "ERROR: CAN'T DELETE NONEXISTENT LINE":GOTO 150
500 IF M=N THEN PRINT "ERROR: PROGRAM BUFFER FULL":GOTO 150
510 REM  MAKE ROOM FOR NEW LINE
520 FOR Y=M TO X STEP -1
530 P(Y+1)=P(Y)
540 P$(Y+1)=P$(Y)
550 NEXT Y
560 REM  STUFF IN NEW LINE NUMBER
570 P(X)=P
580 M=M+1
590 GOTO 650
600 REM      -----SPECIAL CASE: APPEND TO END OF PROGRAM-----
610 IF M=N THEN PRINT "ERROR: PROGRAM BUFFER FULL":GOTO 150
615 IF L=-1 THEN PRINT "ERROR: CAN'T DELETE NONEXISTENT LINE":GOTO 150
620 P(M)=P
625 X=M
630 M=M+1
640 P(M)=10000
650 REM  NOW WE COPY THE REST OF THE INPUT LINE TO THE PROGRAM
660 P$(X)=RIGHT$(C$,LEN(C$)-L+1)
670 GOTO 150
1000 REM
1010 REM      -----'NEW' COMMAND-----
1020 REM
1030 GOSUB 8500
1040 GOTO 150
2000 REM
2010 REM      -----'PLAY' COMMAND-----
2020 REM      MUSIC CODE IN P$ IS "COMPILED" IN MEMORY AND THE
2030 REM      RUNTIME INTERPRETER IS LOADED (WITH RELOCATION).
2040 REM      THEN BASIC CALLS THE MACHINE LANGUAGE INTERPRETER
2050 REM      TO ACTUALLY PLAY THE MUSIC
2060 REM
2065 RESTORE
2070 REM  DEFINE CONSTANTS
2080 K1=2^(1/12)
2090 K6=1.2
2100 REM  COMPUTE BASE FOR COMPILATION
2110 Q=29000: B=Q
2115 PRINT "LOAD ADDRESS=";HEX$(Q)
2120 REM  SET LINE AND CHARACTER NUMBER POINTERS
2130 X=1:Y=1
2140 REM  CHECK IF END OF SOURCE CODE YET
2150 IF Y>LEN(P$(X)) THEN Y=1:X=X+1
2155 IF P(X)=10000 THEN 2890
2160 REM  GET NEXT "WORD" OUT OF P$(X) AT CHARACTER #Y PUT IN W$
2170 REM  SCAN TO NONBLANK
2180 FOR Y1=Y TO LEN(P$(X))
2190 IF MID$(P$(X),Y1,1)<>" " THEN 2230
2200 NEXT Y1
2210 REM  IGNORE BLANK LINES
2220 X=X+1:Y=1:GOTO 2140
2230 REM  COPY INTO W$ UNTIL WE HIT SPACE OR END OF STRING
2240 Y=Y1
2250 W$=""
2260 FOR Y1=Y TO LEN(P$(X))
2270 REM  IF WE HIT A STAR, IGNORE REST OF LINE
2280 IF MID$(P$(X),Y1,1)<>"*" THEN 2290
2282 IF W$="" THEN Y=1:X=X+1:GOTO 2155
2284 Y1=LEN(P$(X))+1:GOTO 2370
2290 REM  IF A SPACE, STOP COPYING
2300 IF MID$(P$(X),Y1,1)=" " THEN 2370
2310 REM  ELSE COPY CHARACTER TO W$
2320 W$=W$+MID$(P$(X),Y1,1)
2330 NEXT Y1
2340 REM  WE HIT END OF STRING- RESET POINTERS TO NEXT LINE
2350 Y=Y1
2360 GOTO 2380
2370 Y=Y1
2380 REM  NOW WE "COMPILE" NOTE IN W$
2390 N1=100:P1=100:T1=100
2400 REM  PROCESS LETTER NAME, SHARPS OR FLATS
2410 C=1:GOSUB 8700
```

simple way to write strings out to the cassette and read them back in). The object of this routine is to save the name of the program, and the program itself (line numbers and text). A % is used as an end-of-file marker. In this particular implementation, the console I/0 routines are POKED so that they talk to the cassette interface, rather than the regular terminal. Thus one can PRINT or INPUT directly from the cassette. CHR$(60) and CHR$(230), used in line 5420, are control codes used by the Tarbell cassette hardware and are invisible to the BASIC program when the tape is read in. When the SAVE operation is performed, the console I/0 routines are patched back to normal.

Lines 6000-6480: The LOAD command. This is implemented in a similar fashion to SAVE. If no file name is given, the first file on the tape is loaded. In this case, the I/0 is modified so that an INPUT is done from the cassette rather than a terminal. You must be careful when using this command that you give the name of a file which you actually have, because the program will keep looking for this file until it is found, and can't be interrupted with a control-c. Further, if you hit RESET on the front panel while the console I/0 is POKEd for the cassette, you still can't recover. The console I/0 must then be patched back to normal manually. If you have a simpler way to save data on a mass-storage device, use it. This is certainly a worst-possible situation.

Lines 7000-7170: The data for loading Microplay, as described for the PLAY command above.

Lines 8000-8080: This routine is used to get the file name from the input line for the LOAD and SAVE commands.

Lines 8500-8580: This subroutine re-initializes the music program area.

Lines 8700-8750: A subroutine used in compiling the music program. This routine gets character #C from W$ if it exists and returns a * if it doesn't.

Lines 9000-9999: This subroutine gets line numbers out of the input line, C$. At entry to this subroutine, L is the character position to start scanning at. P is the line number parameter returned from the subroutine. When this subroutine is exited, L is the character position at which to begin scanning for the *next* parameter, and is -1 if the end of the string is reached. If no parameter is found, P=-1. If the parameter was found but was out of range (line numbers can't be greater than 9999) then E=1, otherwise E=0.

This program compiles the music rather slowly, since it is, after all, written in an interpreted BASIC. However, it does work correctly. This code was written in MITS Extended BASIC, but only to use hexadecimal notation (HEX$ and &H for conversion to hex and hex constants). ■

```
2420 IF X$="A" THEN N1=1
2430 IF X$="B" THEN N1=3
2440 IF X$="C" THEN N1=4
2450 IF X$="D" THEN N1=6
2460 IF X$="E" THEN N1=8
2470 IF X$="F" THEN N1=9
2480 IF X$="G" THEN N1=11
2490 IF N1=100 THEN 2860
2500 C=2:GOSUB 8700
2510 M1=N1
2520 IF X$="!" THEN M1=N1-1:C=C+1
2530 IF X$="#" THEN M1=N1+1:C=C+1
2540 GOSUB 8700
2550 REM  NOW LOOK FOR OCTAVE NUMBER
2560 O=VAL(X$)
2570 IF O<1 OR O>3 THEN 2860
2580 P1=M1+12*(O-1)
2590 C=C+1:GOSUB 8700
2600 REM  LOOK FOR NOTE DURATION
2610 IF X$="S" THEN T1=16
2620 IF X$="E" THEN T1=8
2630 IF X$="Q" THEN T1=4
2640 IF X$="H" THEN T1=2
2650 IF X$="W" THEN T1=1
2660 IF T1=100 THEN 2860
2670 REM  LOOK FOR "." (OPTIONAL)
2680 C=C+1:GOSUB 8700
2690 IF X$="." THEN T1=2*T1/3
2700 REM  CALCULATE CONSTANTS
2710 F1=220*(K1^(P1-1))
2720 T2=10^6/(2*F1)
2730 K3=(T2-56.5)/7.5
2740 K4=F1/(K6*T1)
2750 D3=INT(K4)
2760 D4=2*D3-2*INT(D3/2)
2770 D5=INT(D4/256)
2780 D6=D5+1
2790 D7=D4-D5*256
2800 REM  POKE IN THE PSEUDOCOMPILED CODE
2810 POKE Q,INT(K3+.5)
2820 POKE Q+1,D7
2830 POKE Q+2,D6
2840 Q=Q+3
2845 IF Q+3>M9 THEN 3160
2850 GOTO 2140
2860 REM  COMPILATION ERROR! ABEND! ABEND! ABEND!
2870 PRINT "ERROR IN LINE #";P(X);"NOTE=";W$;" CHARACTER #";C
2880 GOTO 2140
2890 REM  DONE WITH COMPILATION- POKE IN END OF SCORE MARKER
2900 POKE Q,0
2910 REM  USE B1 AS BASE FOR INSERTION OF RUNTIME INTERPRETER
2920 B1=Q+1:R1=INT(B1/256):R2=B1-R1*256
2925 IF B1+110>M9 THEN Q=M9: GOTO 3170
2930 FOR B2=B1 TO B1+110
2940 READ D
2950 IF D<1000 THEN 3030
2960 D=D-1000
2970 REM  RELOCATE THIS AND THE FOLLOWING BYTE
2980 D=D+R2:C6=0:IF D>=256 THEN D=D-256:C6=1
2990 POKE B2,D
3000 B2=B2+1
3010 READ D
3020 D=D+R1+C6
3030 POKE B2,D
3040 NEXT B2
3050 REM  TELL BASIC WHERE WE PUT THE RUNTIME CODE
3060 DEFUSR1=R1*256+R2
3070 PRINT "ENTRY ADDRESS=";HEX$(R1*256+R2)
3080 REM  TELL MUSIC INTERPRETER WHERE THE MUSIC IS
3090 R1=INT(B/256):R2=B-256*R1
3100 POKE B1+4,R2
3110 POKE B1+5,R1
3120 REM  NOW, PLAY THE MUSIC!
3130 X=USR1(0)
3140 REM  AND GO BACK TO COMMAND LEVEL
3150 GOTO 150
3160 REM  RAN OUT OF MEMORY ERROR
3170 PRINT "ERROR: RAN OUT OF MEMORY AT ";HEX$(Q)
3180 PRINT "COMPILATION TERMINATED."
3190 GOTO 150
4000 REM
4010 REM      -----'LIST' COMMAND-----
4020 REM      ALLOWS LIST WITH OR WITHOUT LINE #
4030 REM
4040 IF LEN(C$)=4 THEN 4150
4045 L=5
4050 GOSUB 9000
4060 IF E=1 THEN 4250
4070 IF P=-1 THEN 4150
4080 S1=P
```

```
4090 IF L=-1 THEN 4170
4100 GOSUB 9000
4110 IF E=1 THEN 4250
4120 IF P=-1 THEN 4170
4130 S2=P
4140 GOTO 4180
4150 REM  DEFAULT VALUES FOR S1,S2
4160 S1=1
4170 S2=9999
4180 REM  LIST FROM S1 TO S2
4190 FOR X=1 TO M
4200 IF P(X)<S1 THEN 4230
4210 IF P(X)>S2 THEN 150
4220 PRINT STR$(P(X));P$(X)
4230 NEXT X
4240 REM  ERROR MESSAGE
4250 PRINT "ERROR: BAD LINE NUMBER PARAMETER":GOTO 150
5000 REM
5010 REM      -----'SAVE' COMMAND-----
5020 REM      THE PURPOSE OF THIS ROUTINE IS TO SAVE ARRAYS
5030 REM      P AND P$ WITH NAME N$ ON CASSETTE.  TO ACCOMPLISH
5040 REM      THIS WITH THE TIMING CONSTRAINTS OF THE TARBELL
5050 REM      CASSETTE, WE POKE THE CONSOLE OUTPUT ROUTINE IN BASIC
5060 REM      AND "PRINT" THE DATA ON THE CASSETTE (WITH START
5070 REM      AND SYNC BYTES HOOKED ON FRONT).  A 'Z' INDICATES
5080 REM      END OF FILE.  THIS TECHNIQUE IS DESCRIBED IN THE
5090 REM      TARBELL CASSETTE MANUAL.
5100 REM
5110 REM  GET NAME FROM C$
5120 L=5
5140 POKE &HE02,&H6E
5150 POKE &HE04,&H20
5160 POKE &HE05,&HC2
5170 POKE &HE0A,&H6F
5180 REM  SET UP EOF MARKER
5190 P$(M)="Z"
5200 REM  GET NAME
5210 GOSUB 8000
5220 REM  PUT ON FILE
5230 O$=N$
5240 GOSUB 5400
5250 REM  NOW PUT WHOLE PROGRAM ON FILE
5260 FOR X=1 TO M
5270 O$=STR$(P(X))
5280 GOSUB 5400
5290 O$=P$(X)
5300 GOSUB 5400
5310 NEXT X
5320 REM  NOW, POKE CONSOLE I/O BACK TO NORMAL
5330 POKE &HE02,&H0
5340 POKE &HE04,&H80
5350 POKE &HE05,&HCA
5360 POKE &HE0A,&H1
5370 PRINT "SAVE COMPLETE."
5380 GOTO 150
5390 REM
5400 REM  "PRINT" O$ ON CASSETTE WITH START AND SYNC BYTES
5410 REM
5420 O$=CHR$(60)+CHR$(230)+O$
5430 PRINT O$
5440 REM  TIME DELAY BETWEEN BLOCKS
5450 FOR T0=1 TO 100
5460 NEXT T0
5470 RETURN
6000 REM
6010 REM      -----'LOAD' COMMAND-----
6020 REM      THIS ROUTINE IS IMPLEMENTED IN THE SAME MANNER
6030 REM      AS THE SAVE COMMAND EXCEPT THAT THIS
6040 REM      TIME WE FAKE BASIC INTO USING AN INPUT STATEMENT
6050 REM
6060 GOSUB 8500
6070 REM  CHANGE CONSOLE I/O FOR CASSETTE
6080 REM  MUST ALSO CHANGE OUTPUT TO STOP PRINTING "?"
6090 REM  MUST ALSO CHANGE OUTPUT TO STOP PRINTING "?"
6100 POKE &HE11,&H6E:POKE &HE13,&H10
6110 POKE &HE14,&HC2:POKE &HE18,&H6F
6120 POKE &HE0A,&HFF
6130 S3=0
6150 REM  GET FILE NAME
6160 L=5:GOSUB 8000
6170 REM  NOW WAIT TIL THAT FILE GOES BY...
6180 GOSUB 6450
6190 IF I$=N$ THEN 6240
6200 IF N$="NONE" THEN N$=I$:GOTO 6240
6210 GOSUB 6240
6220 IF I$<>"Z" THEN 6210
6230 GOTO 6120
6240 REM  THIS IS THE FILE WE WANT-- LOAD  IT
6250 FOR X=1 TO N
6260 GOSUB 6450
6270 P(X)=VAL(I$)
6280 GOSUB 6450
6290 P$(X)=" "+I$
6300 IF I$="Z" THEN 6350
6310 NEXT X
6320 S3=1: GOTO 6360
6330 GOSUB 8500
6340 GOTO 150
6350 M=X
6360 REM  POKE CONSOLE I/O ROUTINE BACK TO NORMAL
6370 POKE &HE11,0
6380 POKE &HE13,1
6390 POKE &HE14,&HCA
6400 POKE &HE18,1
6410 POKE &HE0A,1
6420 IF S3=1 THEN PRINT "ERROR: BUFFER TOO SMALL":GOTO 1000
6430 PRINT N$;" LOAD COMPLETE."
6440 GOTO 150
6450 REM  "INPUT" I$ FROM CASSETTE
6460 OUT 110,16
6470 INPUT I$
6480 RETURN
7000 REM
7010 REM      DATA FOR MACHINE LANGUAGE PROGRAM
7020 REM      THESE ARE BYTES LOADED INTO MEMORY
7030 REM      DATA >1000 MEANS THIS AND THE NEXT BYTE
7040 REM      ARE ADDRESSES TO RELOCATE
7050 REM
7060 DATA0,0,0,33,0,1,34,1109,0,42,1109,0
7070 DATA 62,0,190,200,0,0,126,50,1065,0
7080 DATA 50,1096,0,35,126,50,1058,0,50,1093,0
7090 DATA 35,126,50,1059,0,50,1094,0,35,34
7100 DATA 1109,0,205,1051,0,195,1009,0
7110 DATA 33,1101,0,30,8,126,1,0,0,64
7120 DATA 195,1064,0,22,0,211,36,52,53,52,53
7130 DATA 52,53,21,194,1074,0
7140 DATA 174,13,194,1060,0,5,194,1064,0,211,36,35,29
7150 DATA 200,1,0,0,22,0,126,195,1074,0
7160 DATA 149,176,208,254,255,255,176,133
7170 DATA 0,0
8000 REM
8010 REM      GET NAME FROM C$, PUT IT IN N$
8020 REM      STARTS AT CHARACTER POSITION 5
8030 REM
8040 IF LEN(C$)=4 THEN N$="NONE":RETURN
8050 N$=RIGHT$(C$,LEN(C$)-4)
8060 IF LEFT$(N$,1)=" " THEN N$=RIGHT$(N$,LEN(N$)-1):GOTO 8060
8070 IF RIGHT$(N$,1)=" " THEN N$=LEFT$(N$,LEN(N$)-1):GOTO 8070
8080 RETURN
8500 REM
8510 REM      CLEAR PROGRAM AREA
8520 REM
8530 FOR Z6=1 TO N
8540 P(Z6)=0: P$(Z6)=""
8550 NEXT Z6
8560 M=1
8570 P(M)=10000
8580 RETURN
8700 REM
8710 REM      X$=MID$(W$,C,1)
8720 REM      PUT IN DUMMY IF END OF STRING
8730 REM
8740 IF C>LEN(W$) THEN X$="*":RETURN
8750 X$=MID$(W$,C,1):RETURN
9000 REM
9010 REM      GET WHOLE NUMBER SUBROUTINE
9020 REM
9170 E=0
9180 IF L>LEN(C$) THEN L=-1:E=1:RETURN
9190 FOR Z5=L TO LEN(C$)
9200 IF MID$(C$,Z5,1)<>" " THEN 9240
9210 NEXT Z5
9220 REM  NO PARAMETER WAS FOUND
9230 P=-1:L=-1:RETURN
9240 REM  FOUND NONBLANK CHARACTER, IS IT NUMERIC
9250 L=Z5:P=0
9260 A=ASC(MID$(C$,L,1))
9270 IF A<48 OR A>57 THEN 9380
9280 REM  IT IS, SO TACK ON NEW DIGIT
9290 P=10*P+A-48
9300 REM  DO WE HAVE ANY MORE STRING TO PROCESS?
9310 L=L+1:IF L<=LEN(C$) THEN 9260
9320 REM  NO, END OF STRING
9330 L=-1
9340 REM  IS P WITHIN LIMITS
9350 IF P>0 AND P<10000 THEN RETURN
9360 REM  NO IT ISN'T
9370 P=-1:RETURN
9380 REM  COME HERE WHEN WE HIT A NON-NUMERIC AFTER PARAM
9390 IF A<>32 THEN E=1
9400 GOTO 9340
9999 END
```

Everyone's getting personal in Long Beach.

3 full days of technical sessions, exhibits, home-brew displays and the latest on personal and small business computing, all at PERCOMP '78. April 28-29-30.

Just for the fun of it, we have an entire home-brew section... robotics, games, computer music, even every-day, sensible stuff like checkbook balancing and preparing mailing lists. You're sure to take home some new tricks to your computer.

And don't forget, PERCOMP '78 has booth after booth of everything in personal and small business computing.

Attorney Kenneth Widelitz will be on hand with some friendly advice on "Tax Aspects of Lemonaide Stand Computing" while his friend attorney Leonard Tachner delivers the low-down on "Patents, Copyrights and Computers."

Jim Butterfield is on his way from Toronto with the entire, unabridged truth about KIM. Jim co-authored *The First Book of KIM.*

5 months before show time our dynamite exhibit list includes from A to V:

The Astute:
Advanced Computer Products
Alpha Supply Co.
Apple Computer, Inc.
A-Vidd Electronics

Carol Anne Ogdin's keynote address bares the facts on "How Personal Computers Are Being Used Today." Carol comes to us from Software Technique, Inc. in Alexandria, Virginia.

Admission for 3 full days of personal computing, complete with 180 exhibits, 66 fascinating seminars and all the going and coming you want is $10 ($8 for students and juniors) at the door, and $8 ($6 for students and juniors) if you pre-register.

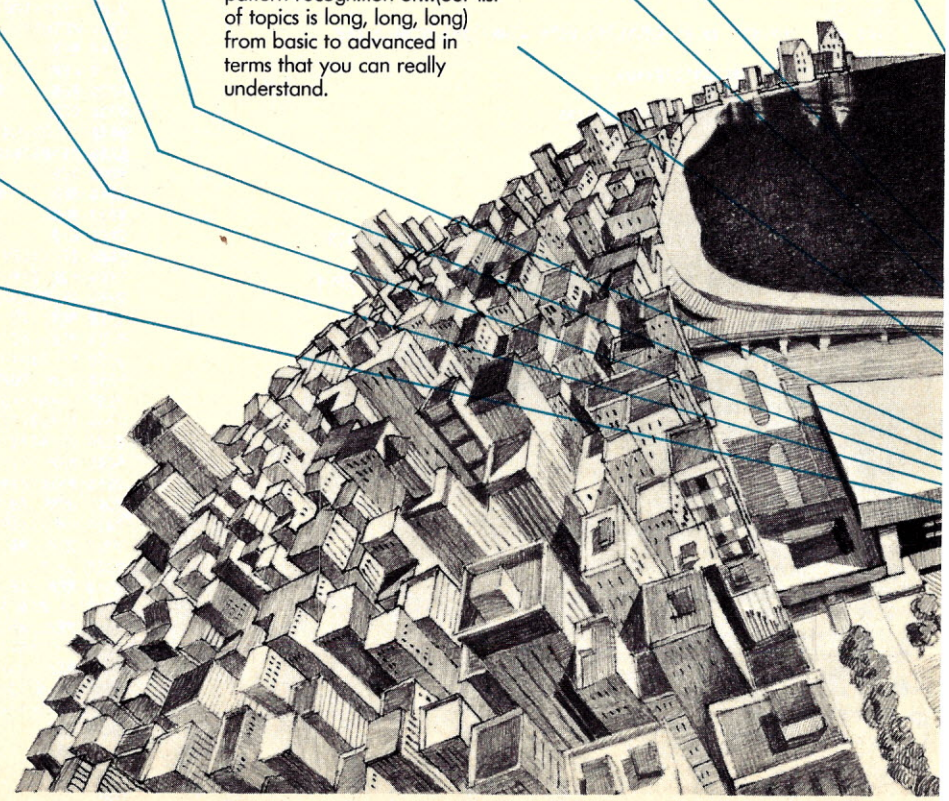
The Brilliant:
Byte Industries Incorporated
Byte Shop Lawndale
Byte Publications, Inc.

Dr. Portia Isaacson, a contributing editor for Datamation and an associate of Byte, brings computer enthusiasts the very latest word on "Computer Store Retailing."

Whether you're just a beginner or a well informed expert, you'll find the latest on ham radio communications, graphic systems, word processing, pattern recognition or...(our list of topics is long, long, long) from basic to advanced in terms that you can really understand.

Louis Field, president of the International Computer Society/ SCCS, gives you everything he's got on "Getting Started in Micro-Computing."

From Creative Computing Magazine comes David Ahl with all you'll ever need to know on "Marketing for the New Manufacturer."



The Captivating:
Calcomp
Computalker Consultants
Computerland
Computer Magazine
The Computer Mart of Orange
Computer Pathways Unlimited, Inc.
Computer Power & Light Inc.
Creative Computing

The Dynamic:
Databyte, Inc.
D.C. Hayes Assoc.
Dilithium Press
Dynabyte, Inc.

The Energetic:
Edwards Assoc.
Electronics Warehouse, Inc.
Electro-Sonic Components, Inc.
Entech

The Hearty:
Heathkit Electronic Centers
Hobby World

The Irresistible:
Interface Age Magazine

The Jovial:
Jade Company
James Henry Co.

The Keen
Kathryn Atwood Enterprises
Kilobaud Magazine

The Magnificent:
Marinchip Systems
Micropolis Corporation
Mission Control

The Omnipotent:
OK Machine & Tool Corp.
Olson Electronics, Inc.
Optical Electronics, Inc.
Orange County Computer Center

The Personable:
Pan Dynamics, Inc.
Personal Computing
Problem Solver Systems, Inc.

The Quintessential:
Quainco Ltd.
Quest Electronics

The Remarkable:
Radio Shack
ROM Publications, Corp.

The Sterling:
SD Sales
Space Byte Corp.
SubLogic Co.
Sunshine Computer Co.
Sybex, Inc.
Szerlip Enterprises

The Tantalizing:
Tandy Computers
Tarbell Electronics
Tech-Mart
Telpar, Inc.
TLF, Corp.

The Ultra:
Ultra-Violet Products, Inc.

The Valiant:
Vector Graphics, Inc.
Vista Computer Co.

RIP THIS COUPON FROM THE PAGE AND GET IT TO US BY APRIL 10.

I want to save time and money. Please send me_____ pre-registration forms.

PERCOMP '78

1833 E. Seventeenth St., Suite 108, Santa Ana, Ca. 92701
(714) 973-0880

Name_____

Address_____

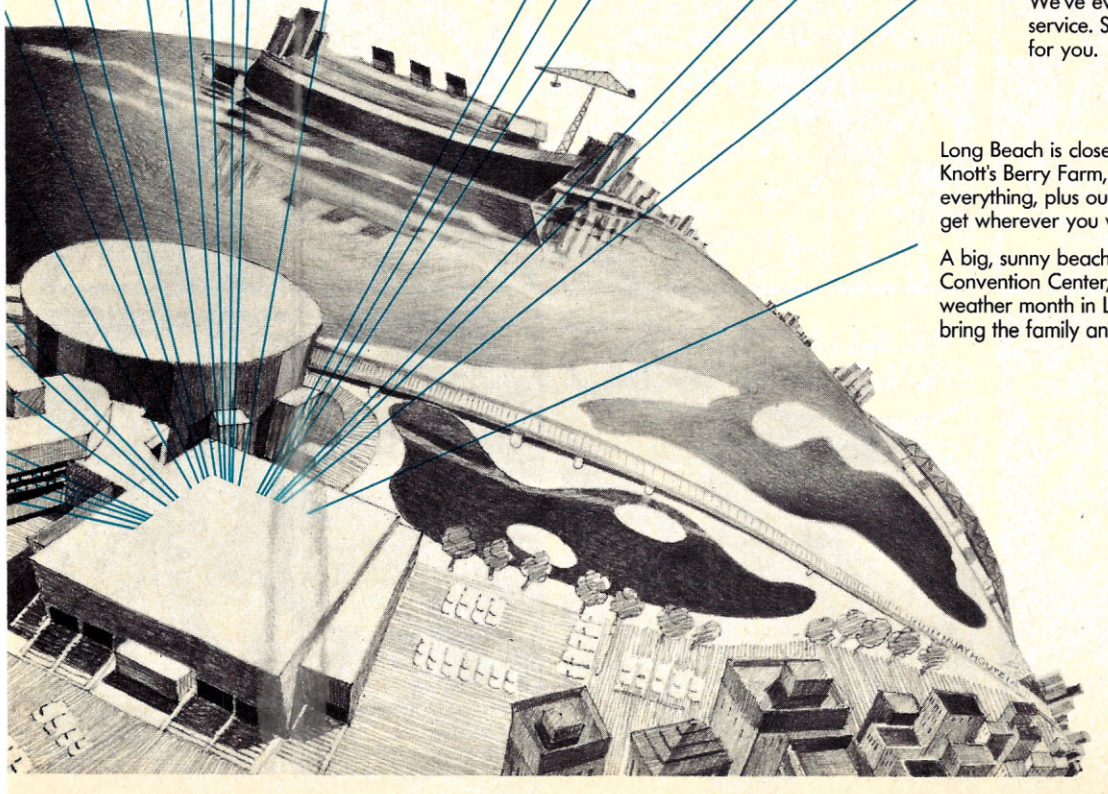
City_____ State_____ Zip_____

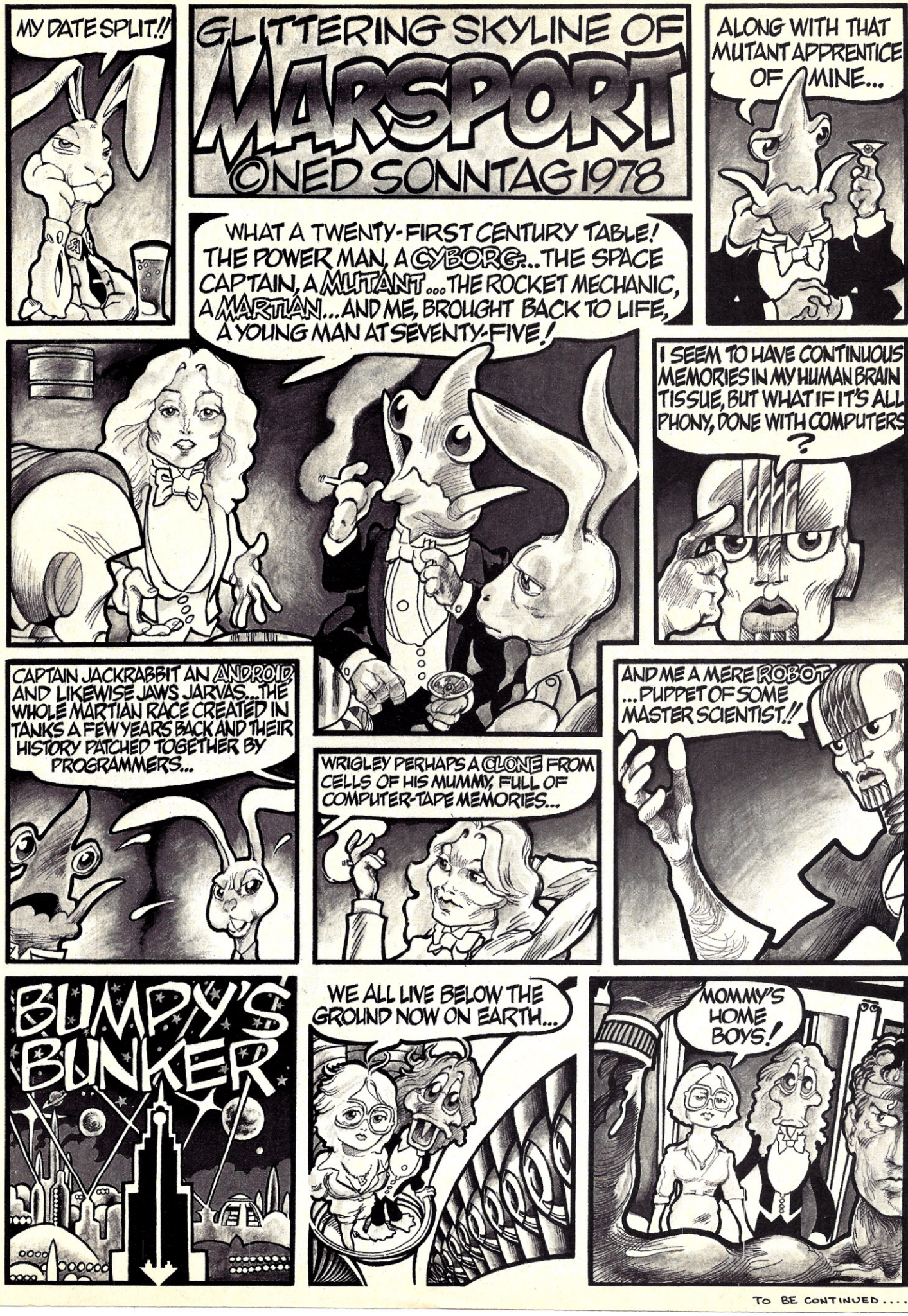
CC-3

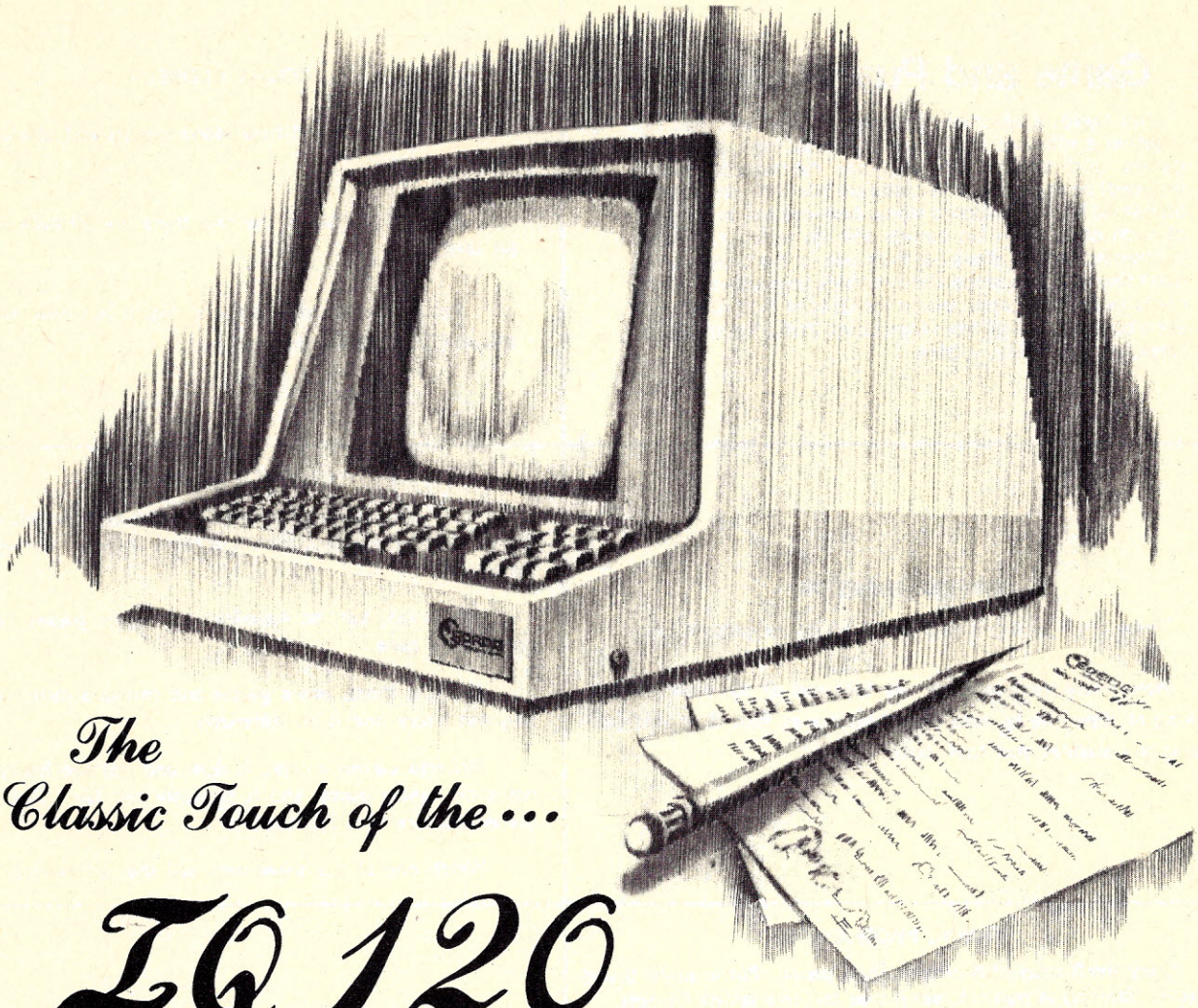
Since everybody's coming, better make your advanced reservations. Pre-register and save (you won't have to wait in line) ...but don't forget about your hotel room. Our staff has reserved rooms in hotels and motels near the Convention Center. We've even arranged for a shuttle bus service. So call and we'll save a room for you.

Long Beach is close to Disneyland, Knott's Berry Farm, Universal Studios... everything, plus our staff will help you get wherever you want to go.

A big, sunny beach is minutes from the Convention Center, and April is a great weather month in Long Beach, so plan to bring the family and have a good time.




MY DATE SPLIT!!
GLITTERING SKYLINE OF MARSPORT
©NED SONNTAG 1978
ALONG WITH THAT MUTANT APPRENTICE OF MINE...
WHAT A TWENTY-FIRST CENTURY TABLE! THE POWER MAN, A CYBORG... THE SPACE CAPTAIN, A MUTANT... THE ROCKET MECHANIC, A MARTIAN... AND ME, BROUGHT BACK TO LIFE, A YOUNG MAN AT SEVENTY-FIVE!
I SEEM TO HAVE CONTINUOUS MEMORIES IN MY HUMAN BRAIN TISSUE, BUT WHAT IF IT'S ALL PHONY, DONE WITH COMPUTERS?
CAPTAIN JACKRABBIT AN ANDROID, AND LIKEWISE JAWS JARVAS... THE WHOLE MARTIAN RACE CREATED IN TANKS A FEW YEARS BACK AND THEIR HISTORY PATCHED TOGETHER BY PROGRAMMERS...
WRIGLEY PERHAPS A CLONE FROM CELLS OF HIS MUMMY, FULL OF COMPUTER-TAPE MEMORIES...
AND ME A MERE ROBOT ...PUPPET OF SOME MASTER SCIENTIST.!!
BUMPY'S BUNKER
WE ALL LIVE BELOW THE GROUND NOW ON EARTH...
MOMMY'S HOME BOYS!
TO BE CONTINUED....



The Classic Touch of the ...

TQ 120

When your customer needs more than just a terminal, ***TQ 120*** offers you an opportunity to deliver craftsmanship and design excellence as well as features not found in similar units.

Features such as upper and lower case / 24 line × 80 characters / address cursor / tabbing / dual intensity / numeric pads / auto repeat / 15 band rates / aux. port and optional printer port.

So if you wish to offer a video display of superior quality, offer the ***TQ 120*** and see how SOROC combines the **"Old World of elegance with the New World of Technology." ... $750.00**

SOROC TECHNOLOGY, INC. 165 FREEDOM AVE., ANAHEIM, CA 92801

800·854-0147

714·992-2860

CIRCLE 41 ON READER SERVICE CARD

Game and Puzzle Cards

Stan Laird, a teacher in Hayward, California has put together a nifty set of 75 cards of math, logic, and word games and puzzles. Several are reproduced (much reduced) on this page. The cards are 10x15 cm and printed on heavy, brightly-colored stock. Answers are on the back of each card. Each set comes with an instruction sheet containing lots of ideas for their use in the classroom. I don't like to guess at the ability of kids these days, but the cards seem most appropriate for Grades 7-10. Card sets are a bargain at $2.50 from Stan Laird, 27948 Farm Hill Dr., Hayward, CA 94542. —DHA

COUNT CLOSELY

A man was asked how many rabbits and chickens he had in his yard.

He said, "Between the two there are 60 eyes and 86 feet."

From what he said, can you figure out how many chickens and rabbits the man had?

MARBLE PROBLEM

There are 4 black, 4 red, and 4 white marbles in a box.

How many marbles must you take out of the box (without looking at them) to be sure that there are at least 2 of one color among the marbles you took out?

WHO WILL SHE MARRY?

A girl once said, "The man I marry will be tall, not dark, quite heavy, will wear glasses and carry a cane. He will be an American."

Jim is tall, fair, an American and wears glasses, but he does not carry a cane.

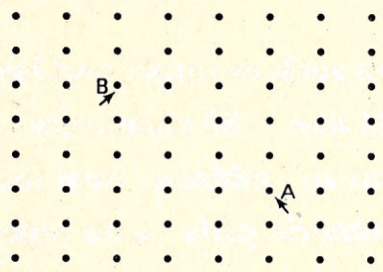
David is short, wears glasses and carries a cane. He is not fair, not heavy and is an American.

Roberto carries a cane, is dark and not too heavy. He is not short, wears glasses and is certainly not fair. Roberto has the appearance of a Spaniard.

Which one of the three men will the girl marry?

PENCIL PROBLEM

Copy the 8 rows of 8 dots on your paper. Put in point B and point A. Starting at point A, see if you can connect all 64 dots with a straight line and end up at point B. **The lines may not cross each other.**



THREE DIGIT NUMBERS

See if you can figure out how many different three digit numbers can be made from the digits 1, 2, 3.

You should have a lot of fun with this one.

BLACK or BLUE

A man who wears either blue or brown socks keeps them all in the same drawer, all mixed up. In total there are 20 blue and 20 brown socks in the drawer. When the man looks for his socks in the morning there is not enough light for him to see the color of the socks.

How many socks must he take out of the drawer to be sure that he has a matching pair?

YOUR ANSWER WILL ALWAYS BE 3

	Example
Think of a number	14
Double it	28
Add 9	37
Subract 3	34
Divide by 2	17
Subtract the original number you thought of	− 14
Your answer will always be	3

Be creative and make your original number a fraction or decimal.

PROGRAMMING FOR EXCELLENCE

WITH TEXTS FROM McGRAW-HILL

BASIC PROGRAMMING FOR COMPUTER LITERACY

David Moursand, University of Oregon 1978, 224 pages, $6.95

This text gives a modern overview of what is involved in working with a computer and writing programs using the BASIC language.

Designed for use in computer literacy instruction, the text avoids the use of higher level mathematics. It begins with problem-solving, then moves on to discuss canned programs and how to read programs, first showing students how to modify existing programs so that they can progress easily and logically to more difficult topics.
Instructor's Manual.

BASIC: A HANDS ON METHOD

Herbert Peckham, Gavilan College 1978, 160 pages, $7.95 softcover

This text teaches BASIC programming skills without going on to higher level mathematics. It introduces the BASIC language, emphasizing programming rather than theory, and is geared to the most popular time sharing computers.
Instructor's Manual.

PROGRAMMING WITH PL/1

Henry Ruston, Polytechnic Institute of New York 1978, 416 pages, $10.50

This comprehensive, highly readable introduction to computer science will activate the student to use the computer frequently and knowledgeably. It introduces the basics of programming with PL/1 very early in the text and presents programming concepts along with the new language, providing a well-rounded grasp of the principles and procedures of modern programming practice.
Instructor's Manual.

A PROGRAMMING PRIMER

Robert P. Taylor, Columbia University 1978, 416 pages, $10.50

A thorough introduction to the concepts and techniques of computer programming, this text focuses on the process of developing problem solutions suitable for computerization. Concepts are presented initially in FPL, then students are shown how to translate these concepts into BASIC, PL/1, COBOL and FORTRAN. Suitable for self-study, the text is useful even where access to computers is extremely limited.

THE ELEMENTS OF PROGRAMMING STYLE, Second Edition

Brian W. Kernighan, and **P.J. Plauger** 1978, 160 pages, $5.95

The second edition of this highly successful text shows how common sense and close attention to style can make any program easier to write, read, and modify over the course of its lifetime.

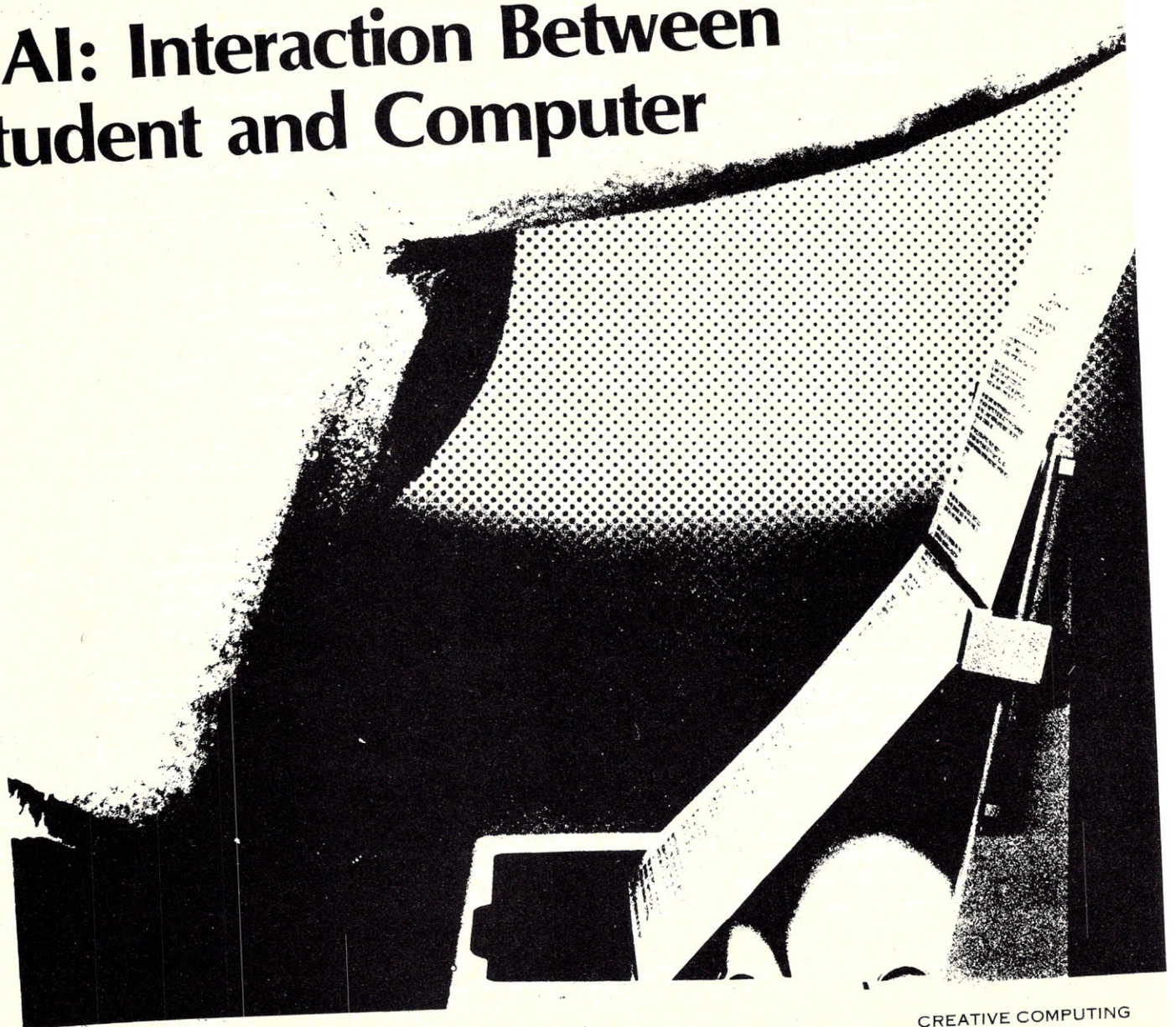
Revisions include increased emphasis on structured programming techniques, more thorough coverage of the principles of good control flow, more material on how to organize large programs, and several new examples and exercises, all carefully checked and tested.

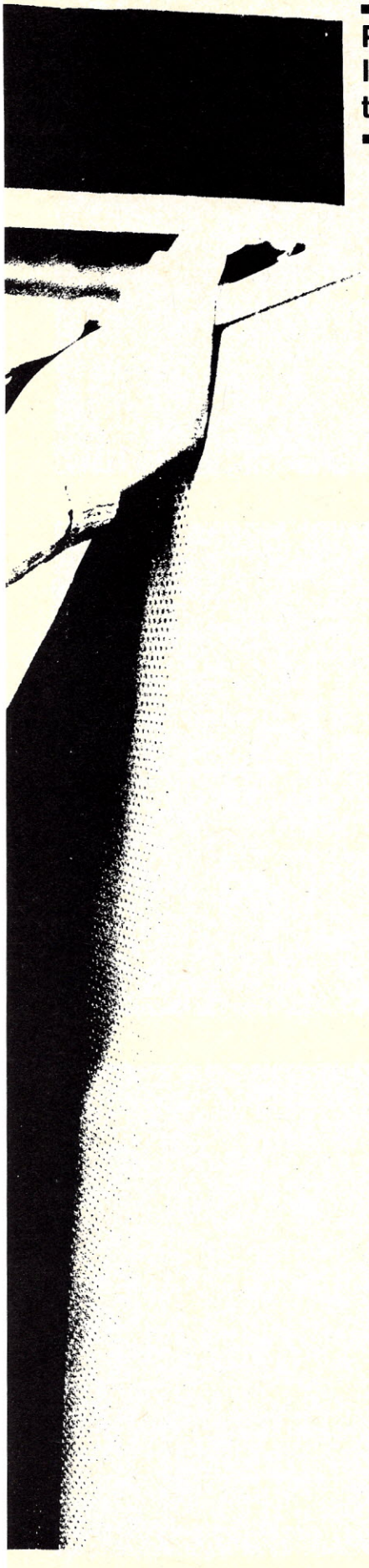
prices subject to change

COLLEGE DIVISION/**McGRAW-HILL BOOK COMPANY**
1221 Avenue of the Americas, New York, N.Y. 10020

CIRCLE 157 ON READER SERVICE CARD

CAI: Interaction Between Student and Computer





Part four in this series on Computer Assisted Instruction investigates "some of the more subtle areas that deal with how the student and computer interact."

Laura L. McLaughlin

The previous article in this series (Nov-Dec 1977, p. 74) gave us an example of a program that would maintain sliding grade levels for a student by problem type, both within a session and from one session to the next. This is a very important concept for good CAI. Now, however, we're going to take a closer look at some of the more subtle areas that deal with how the student and computer interact.

For instance, consider my son Jeff's initial reaction when he sat down at the terminal to try out the program. His first problem was a vertical addition with three numbers of three digits each. "If I could put this in the right direction, it would be easy!", were the first words out of his mouth. And he had a very valid point. Why should he have to put his answer in from left to right just because he's working with a computer?

Another occurrence, in many BASIC programs, is that non-numeric input will either cause the program to abort and return to BASIC (worst case) or generate a non-specific error message which could leave the student confused. Why can't it simply state that what he entered was not a number and it would like him to put his answer in again?

The solution to these problems is simple. The program could do these things — it's just a matter of someone deciding they are important enough to spend some additional programming effort to overcome them (of course, the use of a CRT with cursor control instead of a Teletype might make life a little easier).

Let's consider what kinds of routines we might be able to add to a CAI program to make the interaction between student and computer as smooth as possible, from the student's point of view.

(1) *Editing of input.* This would catch any non-numeric or otherwise invalid input (like, let's not assume on a yes/no answer that if it wasn't a "yes" it was a "no"). We could also give the student an out if he is presented with a problem totally beyond his ability by

Laura L. McLaughlin, Computer Mart of Pennsylvania, 550 DeKalb Pike, King of Prussia, PA 19406.

```
WE ARE GOING TO HELP YOU PRACTICE MULTIPLICATION
WHAT IS YOUR NAME?  JEFF

IF YOU DON'T UNDERSTAND A PROBLEM, OR IT IS TOO DIFFICULT,
JUST TYPE RETURN TO GO ON TO THE NEXT ONE -
  I WILL SHOW YOU THE ANSWER AND GIVE YOU A CHANCE TO STUDY
  IT SO THAT YOU MIGHT BE ABLE TO GET IT THE NEXT TIME

OKAY, JEFF LET'S GET TO IT
THE FIRST SET OF PROBLEMS WILL BE SIMILAR TO THIS:

          727
        X   3
        -----
         2181

YOU HAVE YOUR CHOICE OF ENTERING YOUR ANSWERS FROM
LEFT TO RIGHT (IF YOU WANT TO FIGURE IT OUT IN YOUR HEAD)
OR FROM RIGHT TO LEFT (IF YOU'D RATHER WORK IT OUT STEP
BY STEP).

WOULD YOU LIKE TO ENTER FROM RIGHT TO LEFT
WORKING IT OUT STEP BY STEP? _
```

accepting a CR (carriage return) only as a statement that he has given up. Frustration can be very negative and should therefore be avoided.

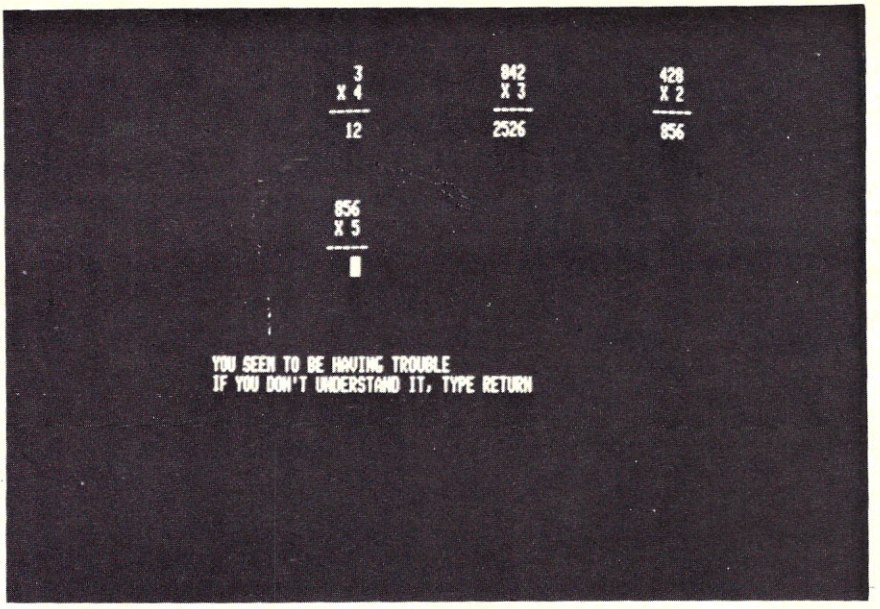
(2) *Answer positioning.* In many cases the student should have the option of entering his answer in the direction that is easiest for him. We could also, in certain cases, allow him to work out the entire problem (including any intermediate steps in long multiplications and divisions, for example). Although at times, for various reasons, we might want to specify these things ourselves.

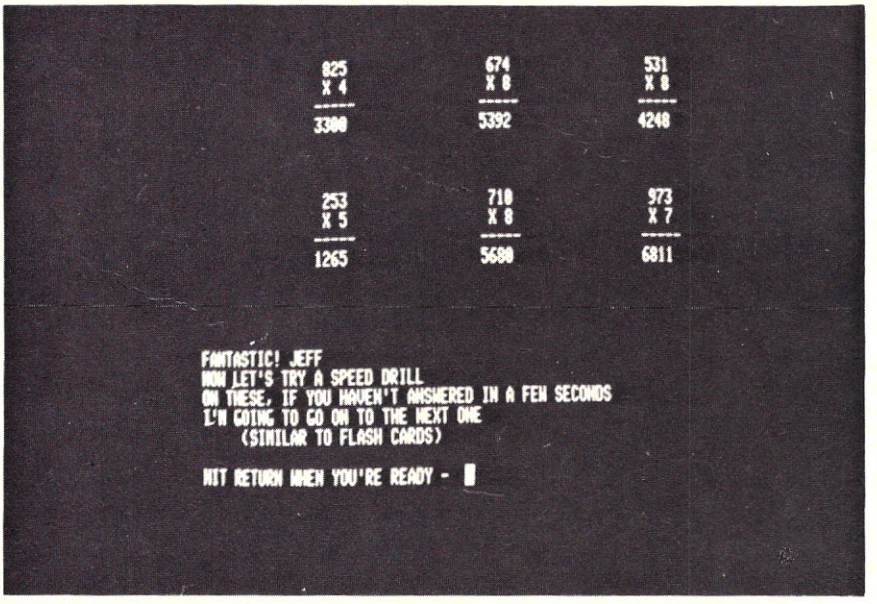
(3) *Timing routines.* The length of time it takes for a response to be entered can be very important. Some students are reluctant to admit they do not understand and will therefore not use the "out" we have given them. So maybe, after a certain period of time, we should prompt them with a message. Another use for a timing routine would be to provide speed drills for things like multiplication tables (automatic flash-cards).

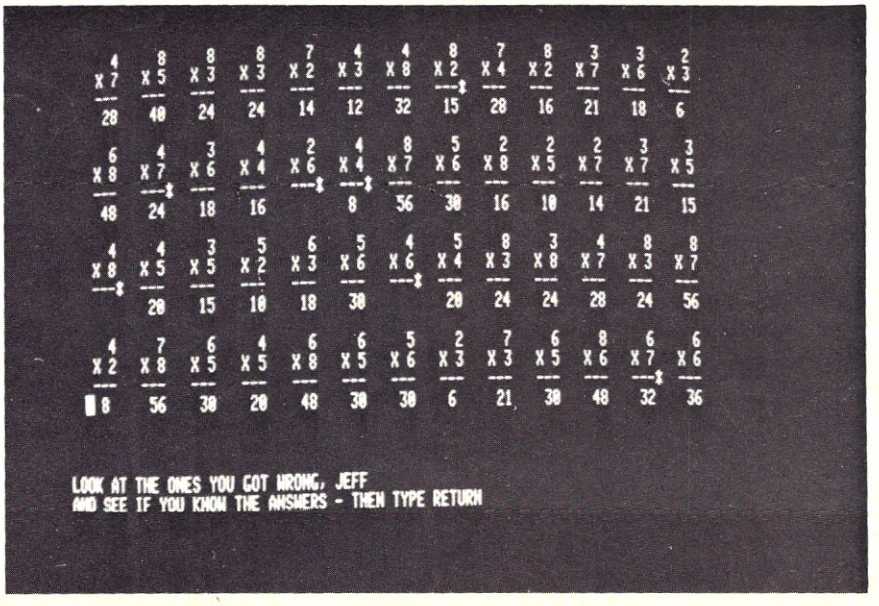
The program included in this article, MATHMULT, which is designed to give a student practice in multiplication, incorporates all of these ideas. Let's take a look at it and see how it is done. Three groups of problems are given to the student, with the code for each starting at lines 100, 200, and 300, respectively.

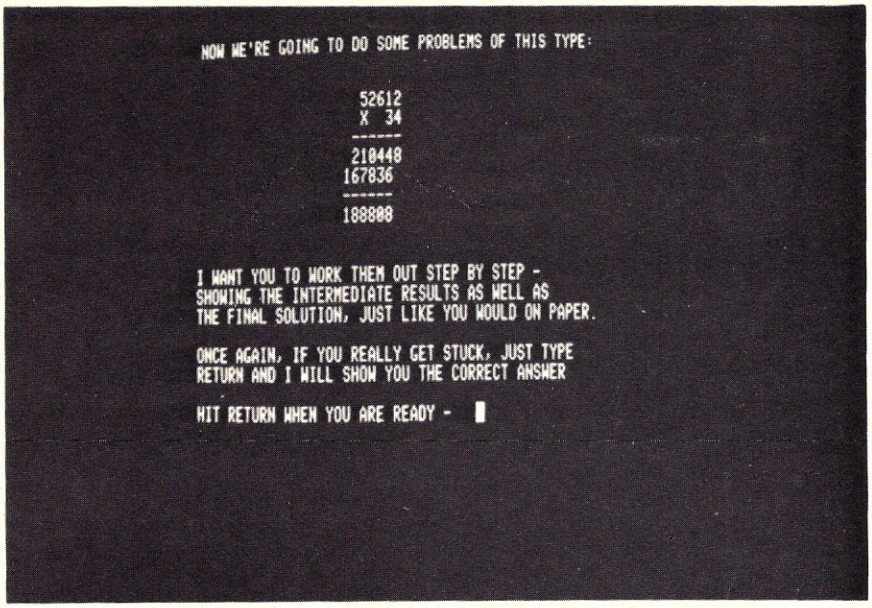
The first set consists of twelve problems that require multiplying a number up to three digits long by a one-digit number. Two screens are presented with six problems on each of them. The student is given a choice between entering his answer from left to right or from right to left. For this level of problem, the option is given to him since for some students it would be easier to do the calculation in their heads while for others it would not. The program will keep track of the time it takes for the student to enter his response and give him a prompting message if he is taking an excessive amount of time. The length of time will be greater if he has elected to enter his answer from left to right, since he must do the entire calculation in his head. The input will be edited to insure that it is numeric. Invalid data is captured immediately on input so it is not even put out to the screen. Instead, the BELL is sounded to indicate that the character entered was not accepted. When he has completed all twelve problems, an appropriate message will be written to indicate how well he did.

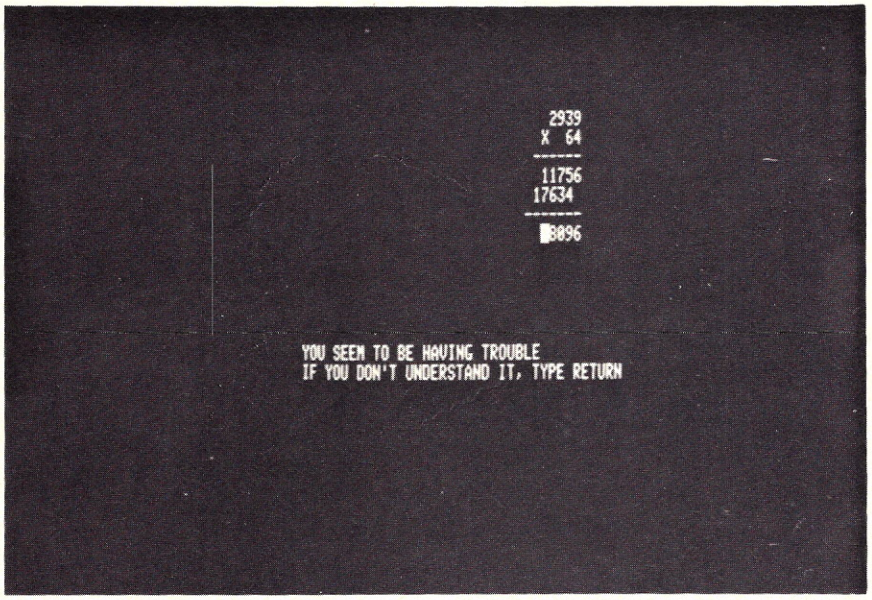
Next the student will be given 52 "flash-card" type problems. They will all be put on one screen, and he will have only a few seconds to enter each response. On these problems, as opposed to the previous ones, if an answer is incorrect, no opportunity is

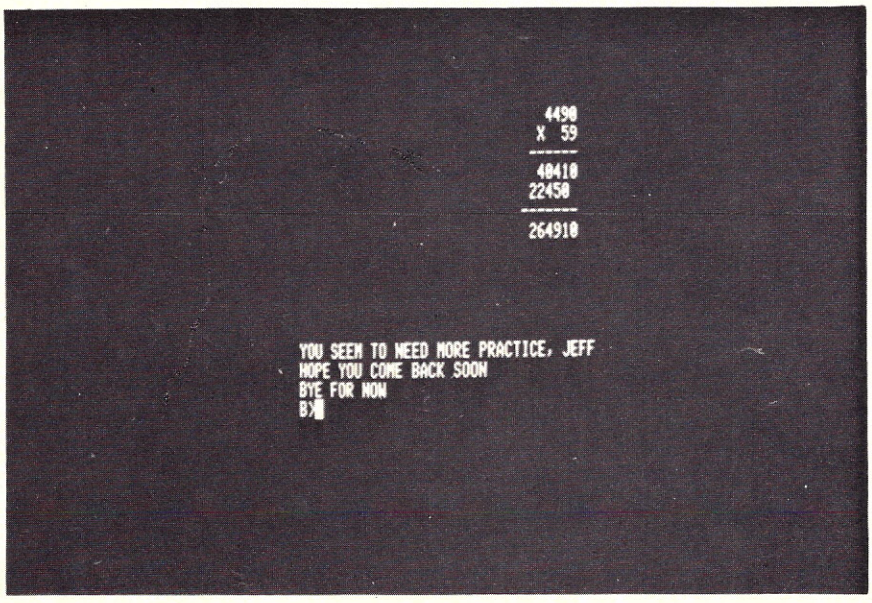












given to re-enter the data since the whole idea is speed. When he is done he will be shown which ones he missed and be given a chance to look them over to see if he knows the answers. When he is ready to go on, the corrected answers will be displayed. This is very important for two reasons: (1) if he simply did not know the right answer it will give him a chance to study it, and (2) it will reinforce his knowledge if he just couldn't get it in the amount of time allowed.

The final set of problems presented will be "long" multiplications: a multiple-digit number times another multiple-digit number. These problems require that the student enter his answers from right to left because we want him to work them out just like he would on paper, including intermediate results. Note that all answers, final as well as intermediate ones, should be edited for non-numerics. In this example, all responses are also checked for a CR only, indicating that the student gave up, and for validity. By providing him with the correct answer on an intermediate result which he cannot get himself, he still has the opportunity to finish the rest of the problem. In other words, all three entries are treated as individual problems, except that only the final answer is actually counted as right or wrong. There are obviously a number of ways to handle this depending on the level of the student. This program is set up for a relative newcomer to this type of problem, and therefore gives him a fair amount of leeway so he has the chance to complete the problems. Remember, this is a practice session, not a test of his knowledge.

This gives us an overall view of what the program is trying to accomplish. It also explains some of the things included for each problem type that will make the student encounter the least amount of frustration in his interaction with the machine. Now let's look at a few of the more general techniques that are incorporated to further increase the student's feeling of comfort with the tool he is using. Many of these points were discussed in the first article of this series. However, to emphasize their importance, we will mention them again. Briefly, some of the considerations should be:

(1) Alignment of numbers
(2) Use of the student's name
(3) Random responses
(4) Informational messages
(5) Variety of layout.

A student can be easily confused if the numbers in a problem are not lined up correctly, causing him to make errors on things he would otherwise be able to do correctly. This program makes use of the PRINT USING

statement to insure the alignment of numbers, a very convenient method if your BASIC has this statement. If it does not, another means for right justification was shown in the previous article.

Personal feedback in the form of using the student's name is very effective. But remember that anything can be overdone. When you are talking to someone, you do not append their name to every statement. Therefore, the program uses the name somewhat selectively.

Which brings us to the use of random responses. Repetition of the same reply can be very annoying. So the program has been set up with a variety of both positive and negative responses which can be selected from randomly, a much better approach than constantly repeating ourselves.

A liberal number of informational messages is displayed throughout a session. This serves a dual purpose: beyond providing the student with an explanation, they tend to break up the session, giving him a chance to take a breath.

There is no reason to always have problems presented in the same format. Using a CRT, why couldn't we sometimes use the center of the screen, or put up multiple problems at the same time. With this in mind, each group of problems in the program is presented with a different layout on the screen. This avoids the monotony of just displaying one problem after another.

This sample program has shown us how many of the important considerations for a good CAI session can be programmed relatively easily. There are, of course, many variations and extensions of these ideas that could be used. For instance, in the "flash-card" section we might want to allow the student a second chance to go back and correct his own mistakes. Or perhaps we'd want to add the code necessary to eliminate duplicate problems.

Ideally, we would like to combine the sliding-grade concept discussed in the previous article with everything we have considered in this one. Moreover, when we did this, we might also like to use the timing routine to supply response time information as further input to our sliding-grade calculations.

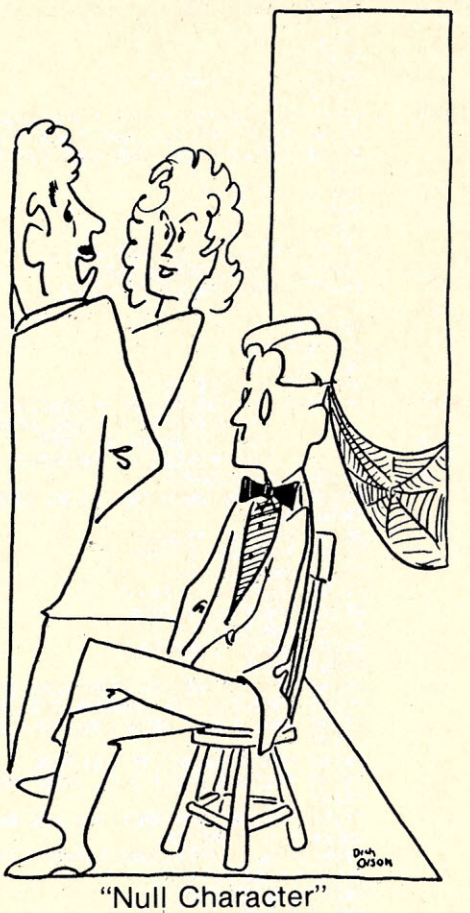
So far, in this series, we have covered much of what is needed to develop a good math drill and practice CAI session. Next we are going to look at some other possibilities for CAI that will still follow the same concepts but which are in areas outside of mathematics. Spelling, grammar, and vocabulary all lend themselves to CAI. And there is no reason why the study of science, social studies, and other subjects cannot be aided by this tool. ■

```
REM     MATHMULT - A MULTIPLICATION DRILL & PRACTICE SESSION
REM             WRITTEN BY LAURA L MCLAUGHLIN  DECEMBER 1977
REM                     COMPUTER MART OF PENNSYLVANIA
REM             USING CBASIC VERSION 1.0
REM                     FOR AN ADM-3A WITH CURSOR CONTROL ENABLED
10      DIM FLASH(4,13)
        FOR I = 1 TO 4
        FOR J = 1 TO 13
        FLASH(I,J)=-1
        NEXT J,I
        CLEAR=26:CR=13:LF=10:BACKSP=8
        ESC=27:DEL=127:BELL=7
        BLANK$="                                                                "
        PRINT CHR$(CLEAR);"WE ARE GOING TO HELP YOU PRACTICE MULTIPLICATION"
        INPUT "WHAT IS YOUR NAME? "; LINE NAME$
        RANDOMIZE
        PRINT
        PRINT "IF YOU DON'T UNDERSTAND A PROBLEM, OR IT IS TOO DIFFICULT,"
        PRINT "JUST TYPE RETURN TO GO ON TO THE NEXT ONE -"
        PRINT "  I WILL SHOW YOU THE ANSWER AND GIVE YOU A CHANCE TO STUDY"
        PRINT "  IT SO THAT YOU MIGHT BE ABLE TO GET IT THE NEXT TIME"
        PRINT
        PRINT "OKAY, ";NAME$;" LET'S GET TO IT"
REM     GROUP 1: 12 PROBLEMS OF 3 DIGITS BY 1 DIGIT - SIX TO A SCREEN
REM             STUDENT OPTION ON DIRECTION OF ENTRY
REM             TWO RETRIES ON INCORRECT ANSWER
100     PRINT "THE FIRST SET OF PROBLEMS WILL BE SIMILAR TO THIS:"
        PRINT
        PRINT TAB(26);"727"
        PRINT TAB(26);"X 3"
        PRINT TAB(25);"------"
        PRINT TAB(25);"2181"
        PRINT
        PRINT "YOU HAVE YOUR CHOICE OF ENTERING YOUR ANSWERS FROM"
        PRINT "LEFT TO RIGHT (IF YOU WANT TO FIGURE IT OUT IN YOUR HEAD)"
        PRINT "OR FROM RIGHT TO LEFT (IF YOU'D RATHER WORK IT OUT STEP"
        PRINT "BY STEP)."
        PRINT
        PRINT "WOULD YOU LIKE TO ENTER FROM RIGHT TO LEFT"
        INPUT "WORKING IT OUT STEP BY STEP? ";A$
100.1   IF LEFT$(A$,1)="Y" THEN DIREC$="RL":LIMIT=1500:GOTO 100.3
        IF LEFT$(A$,1)="N" THEN DIREC$="LR":LIMIT=3000:GOTO 100.3
        PRINT "I DON'T UNDERSTAND, PLEASE ANSWER YES OR NO ";
        INPUT A$
        GOTO 100.1
100.3   TYPE$="REG":NUM1=998:NUM2=7:NUM3=2
        FOR I = 1 TO 2
        X=15:Y=2
        PRINT CHR$(CLEAR);
        FOR J = 1 TO 2
        FOR K = 1 TO 3
        GOSUB 900.1
        Y1=Y:X1=X+1:GOSUB 900.5
        PRINT USING "###";A
        Y1=Y+1:X1=X+1:GOSUB 900.5
        PRINT USING "X #";B
        Y1=Y+2:X1=X:GOSUB 900.5:PRINT "-----"
        Y1=Y+3:WRONG=0
        IF DIREC$="LR" THEN X1=X ELSE X1=X+3
        GOSUB 400
        X=X+20
        NEXT K
        Y=Y+7:X=15
        NEXT J,I
        GOSUB 900.2
        IF RIGHT=12 THEN PRINT "FANTASTIC! ";NAME$:GOTO 200
        IF RIGHT<8 THEN PRINT "LOOKS LIKE YOU COULD USE MORE PRACTICE":GOTO 200
        PRINT "THAT WAS PRETTY GOOD ";NAME$
REM     GROUP 2: 52 PROBLEMS OF 1 DIGIT BY 1 DIGIT - ALL ON ONE SCREEN
REM             ENTRY - LEFT TO RIGHT
REM             TIME OUT AFTER A FEW SECONDS
REM             NO RETRIES ALLOWED - ERRORS & CORRECTIONS SHOWN AT END
200     PRINT "NOW LET'S TRY A SPEED DRILL"
        PRINT "ON THESE, IF YOU HAVEN'T ANSWERED IN A FEW SECONDS"
        PRINT "I'M GOING TO GO ON TO THE NEXT ONE"
        PRINT "      (SIMILAR TO FLASH CARDS)"
        PRINT
        PRINT "HIT RETURN WHEN YOU'RE READY -";:INPUT " ";LINE A$
        DIREC$="LR":TYPE$="FLASH":NUM1=7:NUM3=2:LIMIT=250:RIGHT=0
        Y=1:PRINT CHR$(CLEAR);
        FOR I = 1 TO 4
        X=4
        FOR J = 1 TO 13
        GOSUB 900.1
        Y1=Y:X1=X+1:GOSUB 900.5:PRINT A
        Y1=Y+1:X1=X:GOSUB 900.5:PRINT "X";B
        Y1=Y+2:X1=X:GOSUB 900.5:PRINT "---"
        Y1=Y+3:X1=X+1
        GOSUB 400
        X=X+6
        NEXT J
        Y=Y+5
        NEXT I
        GOSUB 900.2
        IF RIGHT = 52 THEN PRINT "YOU GOT THEM ALL RIGHT!":GOTO 300
REM     LOOP FOR SHOWING MISTAKES
        PRINT "LOOK AT THE ONES YOU GOT WRONG, ";NAME$
        PRINT "AND SEE IF YOU KNOW THE ANSWERS - THEN TYPE RETURN ";
        Y=1
        FOR I = 1 TO 4
        X=4
```

```
        FOR J = 1 TO 13
        IF FLASH(I,J) = -1 THEN 200.5
        Y1=Y+2:X1=X+3:GOSUB 900.5:PRINT "*"
200.5   X=X+6
        NEXT J
        Y=Y+5
        NEXT I
        INPUT " ";  LINE A$
        GOSUB 900.2
REM     LOOP FOR SHOWING CORRECT ANSWERS
        PRINT "HERE ARE ALL THE CORRECT ANSWERS"
        PRINT "STUDY THEM, AND THEN HIT RETURN TO GO ON ";NAME$;
        Y=1
        FOR I = 1 TO 4
        X=4
        FOR J = 1 TO 13
        IF FLASH(I,J)= -1 THEN 200.9
        Y1=Y+3:X1=X+1:GOSUB 900.5
        PRINT USING "##";FLASH(I,J)
200.9   X=X+6
        NEXT J
        Y=Y+5:PRINT
        NEXT I
        INPUT " "; LINE A$
REM     GROUP 3: 4 PROBLEMS OF 4 DIGITS BY 2 DIGITS - ONE PER SCREEN
REM             INTERMEDIATE & FINAL RESULTS TO BE ENTERED
REM             ENTRY - RIGHT TO LEFT
REM             TWO RETRIES ON INCORRECT ANSWER FOR BOTH INTERMEDIATES & FINAL
300     PRINT CHR$(CLEAR);
        PRINT "NOW WE'RE GOING TO DO SOME PROBLEMS OF THIS TYPE:"
        PRINT:PRINT
        PRINT TAB(21);"52612"
        PRINT TAB(21);"X  34"
        PRINT TAB(20);"------"
        PRINT TAB(20);"210448"
        PRINT TAB(19);"167836"
        PRINT TAB(19);"------"
        PRINT TAB(19);"188808"
        PRINT:PRINT
        PRINT "I WANT YOU TO WORK THEM OUT STEP BY STEP -"
        PRINT "SHOWING THE INTERMEDIATE RESULTS AS WELL AS"
        PRINT "THE FINAL SOLUTION, JUST LIKE YOU WOULD ON PAPER."
        PRINT
        PRINT "ONCE AGAIN, IF YOU REALLY GET STUCK, JUST TYPE"
        PRINT "RETURN AND I WILL SHOW YOU THE CORRECT ANSWER"
        PRINT
        PRINT "HIT RETURN WHEN YOU ARE READY - ";
        INPUT " "; LINE A$
        DIREC$="RL":LIMIT=2000:TYPE$="DIFF":NUM1=9988:NUM2=88:NUM3=11
        X=30:Y=5:RIGHT=0
        FOR I = 1 TO 2
        FOR J = 1 TO 2
        GOSUB 900.1
        PRINT CHR$(CLEAR);
        Y1=Y:X1=X:GOSUB 900.5
        PRINT USING "#####";A
        Y1=Y+1:X1=X:GOSUB 900.5
        PRINT USING "X  ##";B
        Y1=Y+2:X1=X-1:GOSUB 900.5:PRINT "------"
        Y1=Y+3:X1=X+4:WRONG=0
        LEVEL=1:TYPE$="INTERMEDIATE":GOSUB 400
        Y1=Y+4:X1=X+3:WRONG=0
        LEVEL=2:GOSUB 400
        Y1=Y+5:X1=X-2:GOSUB 900.5:PRINT "-------"
        Y1=Y+6:X1=X+4:WRONG=0
        LEVEL=3:TYPE$="DIFF":GOSUB 400
        NEXT J,I
        GOSUB 900.2
        IF RIGHT=4 THEN PRINT NAME$;" THAT WAS EXCELLENT":GOTO 300.9
        IF RIGHT>1 THEN PRINT "NOT TOO BAD, ";NAME$:GOTO 300.9
        PRINT "YOU SEEM TO NEED MORE PRACTICE, ";NAME$
REM     END OF SESSION
300.9   PRINT "HOPE YOU COME BACK SOON"
        PRINT "BYE FOR NOW":STOP
REM     VALIDATE ANSWER ROUTINE
REM             CHECKS FOR NO ANSWER (CR ONLY) FIRST
REM               THEN CHECKS IF RIGHT OR WRONG (AND COUNTS EACH)
400     GOSUB 900.6:GOSUB 500
        IF LEN(ANSWER$) = 0 THEN 450
        ANSWER=VAL(ANSWER$)
        IF (TYPE$="INTERMEDIATE") AND (LEVEL=1) AND (ANSWER=ANSW1) THEN RETURN
        IF (TYPE$="INTERMEDIATE") AND (LEVEL=2) AND (ANSWER=ANSW2) THEN RETURN
        IF TYPE$="INTERMEDIATE" THEN 400.07
        IF ANSWER <> ANSW THEN 400.07
        RIGHT=RIGHT+1
REM     SELECTS RANDOM RESPONSE FOR CORRECT ANSWER (EXCEPT FOR FLASH CARDS)
REM             RETURNS TO GET NEXT PROBLEM
        IF TYPE$ = "FLASH" THEN RETURN
        GOSUB 900.2
        ON (RND*5)+1 GOTO 400.051, 400.052, 400.053, 400.054, 400.055, 400.056
400.051 PRINT "THAT'S RIGHT, ";NAME$:GOSUB 900.7:RETURN
400.052 PRINT "VERY GOOD":GOSUB 900.7:RETURN
400.053 PRINT "KEEP UP THE GOOD WORK ";NAME$:GOSUB 900.7:RETURN
400.054 PRINT "YOU GOT IT":GOSUB 900.7:RETURN
400.055 PRINT "PERFECT ";NAME$:GOSUB 900.7:RETURN
400.056 PRINT "THAT'S CORRECT":GOSUB 900.7:RETURN
REM     SELECTS RANDOM RESPONSE FOR WRONG ANSWER (EXCEPT FOR FLASH CARD)
REM             IF WRONG MORE THAN TWICE BRANCHES TO GIVE ANSWER ROUTINE
REM               OTHERWISE RETURNS TO GET NEW ANSWER
400.07  IF TYPE$ = "FLASH" THEN FLASH(I,J)=ANSW:RETURN
        WRONG = WRONG + 1
```

```
        GOSUB 900.2
        IF WRONG > 2 THEN 450
        ON (RND*5)+1 GOTO 400.071, 400.072, 400.073, 400.074, 400.075, 400.076
400.071 PRINT "SORRY, ";NAME$;" TRY AGAIN":GOTO 400
400.072 PRINT "THAT'S NOT IT - ONCE MORE, OK":GOTO 400
400.073 PRINT "TRY AGAIN, ";NAME$:GOTO 400
400.074 PRINT "YOU CAN GET IT, TRY AGAIN":GOTO 400
400.075 PRINT "NOPE, ";NAME$;" ONE MORE TIME":GOTO 400
400.076 PRINT "THINK ABOUT THAT AGAIN":GOTO 400
REM     GIVE ANSWER ROUTINE
REM             RETURNS TO GET NEXT PROBLEM
450     IF TYPE$="FLASH" THEN FLASH(I,J)=ANSW: RETURN
        GOSUB 900.2
        PRINT "I GUESS THIS ONE'S A LITTLE TOUGH FOR YOU"
        PRINT "I'LL TELL YOU THE ANSWER AND YOU TRY TO FIGURE IT OUT"
        PRINT "WHEN YOU'RE READY TO CONTINUE, TYPE RETURN "
        IF DIREC$="RL" THEN X1=X1-5
        GOSUB 900.5
        IF TYPE$="INTERMEDIATE" AND LEVEL=1 THEN PRINT USING "######"; ANSW1
        IF TYPE$="INTERMEDIATE" AND LEVEL=2 THEN PRINT USING "######"; ANSW2
        IF TYPE$<>"INTERMEDIATE"  THEN PRINT USING "######"; ANSW
        INPUT " "; LINE A$
        RETURN
REM     GET ANSWER ROUTINE
REM             CHECKS TIME LIMIT & PROMPTS IF EXCEEDED (EXCEPT FOR FLASH CARD)
REM     INPUTS ANSWER DIGIT BY DIGIT MAINTAINING CURSOR POSITIONING
REM             CHECKS FOR CR - TO INDICATE ENTRY COMPLETE
REM             CHECKS FOR DELETE CHAR
REM             CHECKS FOR NON-NUMERIC - DOES NOT PUT IT OUT, BUT RINGS BELL
REM     BUILDS ANSWER IN A WORK STRING
500     WORK$ = ""
        XSAVE=X1
500.01  GOSUB 900.3
        IF TIME = LIMIT AND TYPE$ = "FLASH" THEN 500.09
        IF TIME = LIMIT THEN \
                GOSUB 900.2:\
                PRINT "YOU SEEM TO BE HAVING TROUBLE":\
                PRINT "IF YOU DON'T UNDERSTAND IT, TYPE RETURN":\
                GOSUB 900.6:\
                GOTO 500.01
        DIGIT = (INP(2) AND 127)
        DIGIT$ = CHR$(DIGIT)
        IF DIGIT$=CHR$(CR) THEN PRINT CHR$(LF);CHR$(CR):GOTO 500.09
        IF DIGIT$=CHR$(DEL) OR DIGIT$="_" THEN GOSUB 900.4:GOTO 500.01
        NUMERIC=MATCH("#",DIGIT$,1)
        IF NUMERIC <> 1 THEN DIGIT=BELL:GOSUB 900.8:GOTO 500.01
        DIGIT=ASC(DIGIT$):GOSUB 900.8
        IF DIREC$="LR" THEN WORK$=WORK$+DIGIT$:X1=X1+1:GOTO 500.01
        DIGIT=BACKSP:GOSUB 900.8:GOSUB 900.8:WORK$=DIGIT$+WORK$:X1=X1-1
        GOTO 500.01
500.09  IF XSAVE <> X1 THEN X1=XSAVE:ANSWER$ = WORK$ ELSE ANSWER$=""
        RETURN
REM     GET RANDOM NUMBERS ROUTINE
REM             ALSO SETS CORRECT ANSWER VARIABLES
900.1   A=INT(RND*NUM1)+NUM3
        B=INT(RND*NUM2)+NUM3
        ANSW=A*B
        IF TYPE$="REG" OR TYPE$="FLASH" THEN RETURN
        B$=STR$(B)
        B1=VAL(RIGHT$(B$,1)):B2=VAL(LEFT$(B$,1))
        ANSW1=A*B1:ANSW2=A*B2
        RETURN
REM     BLANK BOTTOM OF SCREEN ROUTINE
900.2   IF TYPE$="FLASH" THEN ROW=52 ELSE ROW=48
        COL=32
        PRINT CHR$(ESC);"=";CHR$(ROW);CHR$(COL);
        PRINT BLANK$:PRINT BLANK$:PRINT BLANK$
        PRINT CHR$(ESC);"=";CHR$(ROW);CHR$(COL);
        RETURN
REM     TIMING ROUTINE
REM             CHECKS IF ANY DATA HAS BEEN ENTERED
REM                TIMING OUT WHEN LIMIT REACHED
900.3   TIME=0
900.31  IF (INP(3) AND 2) = 2 THEN RETURN
        TIME=TIME+1
        IF TIME < LIMIT THEN 900.31
        RETURN
REM     DELETE CHAR ROUTINE
REM             DIRECTION OF ANSWER ENTRY ACCOUNTED FOR
900.4   IF DIREC$ = "LR" THEN\
                DIGIT=BACKSP:GOSUB 900.8:\
                DIGIT=ASC(" "):GOSUB 900.8:\
                DIGIT=BACKSP:GOSUB 900.8:\
                WORK$=LEFT$(WORK$,LEN(WORK$)-1):\
                RETURN
        DIGIT=ASC(" "):GOSUB 900.8:GOSUB 900.8
        DIGIT=BACKSP:GOSUB 900.8
        WORK$=RIGHT$(WORK$,LEN(WORK$)-1)
        RETURN
REM     POSITION CURSOR ROUTINE - IF NEXT OUTPUT USES PRINT STATEMENT
REM             Y1 = # LINES DOWN : X1 = # CHARS OVER
900.5   POS$=CHR$(ESC)+"="+CHR$(Y1+31)+CHR$(X1+31)
        PRINT POS$;
        RETURN
REM     POSITION CURSOR ROUTINE - IF NEXT OUTPUT USES OUT CHAR ROUTINE
REM             Y1 = # LINES DOWN : X1 = # CHARS OVER
900.6   DIGIT=ESC:GOSUB 900.8
        DIGIT=ASC("="):GOSUB 900.8
        DIGIT=Y1+31:GOSUB 900.8
        DIGIT=X1+31:GOSUB 900.8
        RETURN
REM     WAIT ROUTINE
REM             PROVIDES DELAY LOOP BEFORE GOING ON TO NEXT PROBLEM
900.7   FOR WAIT = 1 TO 500
        NEXT WAIT
        RETURN
REM     OUT CHAR ROUTINE
REM             PUTS OUT 1 CHAR DIRECTLY TO TERMINAL OUTPUT PORT
900.8   IF (INP(3) AND 4) <> 4 THEN 900.8
        OUT 2,DIGIT
        RETURN
        END
```

D.P. DOODLES



"Null Character"

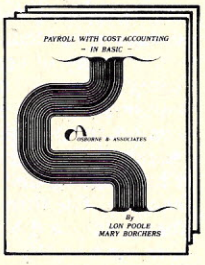
OSBORNE & ASSOCIATES, INC.

The World Leaders In Microprocessor Books

If you want information on microprocessors, begin with the Osborne books.

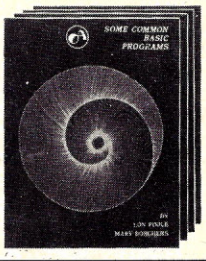
PROGRAM BOOKS WRITTEN IN BASIC

Payroll With Cost Accounting
Accounts Payable And Accounts Receivable
General Ledger



These books may be used independently, or implemented together as a complete accounting system. Each contains program listings, user's manual and thorough documentation. Written in an extended version of BASIC.
#22002 (400 pages), #23002*, #24002*

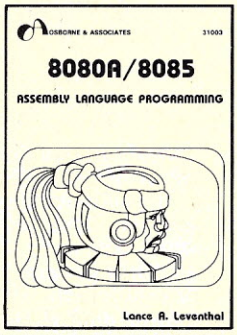
Some Common BASIC Programs



76 short practical programs, most of which can be used on any microcomputer with any version of BASIC. Complete with program descriptions, listings, remarks and examples.
#21002 (200 pages)

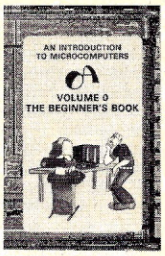
ASSEMBLY LANGUAGE PROGRAMMING

8080A/8085 Assembly Language Programming
6800 Assembly Language Programming



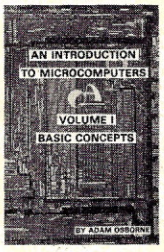
Assembly language primers in the classical sense, these books treat assembly language as a means of programming a microcomputer. Full of simple programming examples.
#31003 (400 pages), 32003*

AN INTRODUCTION TO MICROCOMPUTERS



Volume 0 - The Beginner's Book

If you know nothing about computers, then this is the book with which to start. It tells what computers are all about and describes their component parts. When you've read this, you'll be ready for Volume I.
#6001 (300 pages)



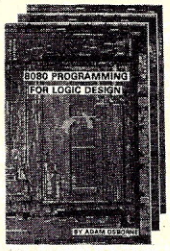
Volume I — Basic Concepts

The world's best selling computer text describes those characteristics common to all microprocessors yet specific to none. All hardware and programming concepts you will ever need to know are explained at an elementary level.
#2001 (350 pages)

Volume II — Some Real Products (revised June 1977)

Every common microprocessor and all support devices are described. Only data sheets are copied from manufacturers. Major chip slice products are also discussed.
#3001A (1250 pages)

PROGRAMMING FOR LOGIC DESIGN



8080 Programming For Logic Design
6800 Programming For Logic Design
Z80 Programming For Logic Design

These books describe the meeting ground of programmers and logic designers; written for both, they provide detailed examples to illustrate effective usage of microprocessors in traditional digital applications.
#4001, #5001, #7001 (300 pages each)

OSBORNE & ASSOCIATES, INC. • P.O. Box 2036 • Berkeley, California 94702 • DEPT. L2

		PRICE	QTY	AMT
6001	Volume 0 — The Beginner's Book	$ 7.50		
2001	Volume I — Basic Concepts	7.50		
3001A	Volume II — Some Real Products	15.00		
4001	8080 Programming For Logic Design	7.50		
5001	6800 Programming For Logic Design	7.50		
7001	Z80 Programming For Logic Design	7.50		
31003	8080A/8085 Assembly Language Programming	7.50		
21002	Some Common BASIC Programs	7.50		
22002	Payroll With Cost Accounting	12.50		
			TOTAL	
			Sales Tax (Calif. residents only)	
			Shipping Charges	
			TOTAL AMOUNT OF PURCHASE	

- 6-1/2%, SF Bay Area residents only
- 6%, California residents outside SF Bay Area
- Payment must be enclosed for orders of 10 books or less.
- I have enclosed:

☐ check ☐ money order

NAME

ADDRESS

CITY STATE ZIP PHONE

SHIPPING CHARGES Shipping charges for bulk orders to be arranged.

☐ 4th class (no charge, allow 3-4 weeks within USA, not applicable to discounted orders)

☐ $.50 per book, UPS (allow 10 days) in the U.S.

☐ $1.50 per book, special rush shipment by air in the U.S.

☐ $3.00 per book, foreign air mail

*This book is not yet available.
Please notify me when it is published:

☐ 23002 Accounts Payable and Accounts Receivable
☐ 24002 General Ledger
☐ 32003 6800 Assembly Language Programming

Please send information on:

☐ consignment
☐ dealer discount
☐ foreign distributors

F3

CIRCLE 118 ON READER SERVICE CARD

. . . reviews

Computer Science For The Teacher. James L. Poirot and David N. Groves. vii + 262 pp. 6" x 9", paper. Sterling Swift Publishing Company, P.O. Box 188, Manchaca, Texas 78652, $8.95. 1976.

As the name of this excellent book suggests, the contents have been selected by the authors to help those classroom teachers (or future classroom teachers) who find themselves trying to use the computer in secondary education.

A history of computers, the major componentes of a digital computer, and the difference between digital and analog computers lead off the text. Two chapters follow, one on computer arithmetic, and the other on logic and Boolean Algebra. They also include a look at the half and full adders. Flowcharting and its use in problem-solving comprise a fourth chapter. The fifth chapter consists of a look at a programming language (via BASIC), and some comparisons between BASIC and Fortran IV. The next two chapters concern themselves with the uses of computers in instructional assistance (grade analysis, drill work, CAI, etc.), and in administrative assistance (attendance, grade reporting, scheduling, payroll, etc.). The final chapter in the book looks at computer games and simulations, and their educational and motivational uses.

Each topic presented with a list of student-oriented objectives and suggestions for presentation of the material. Examples and exercise sets (with answers for selected exercises) complement each other and the text content. Most chapters include a list of references and suggested sources for additional material.

Every teacher involved in computer education should have a copy of this book. Teacher-preparatory institutions should be encouraged to consider this as a possible text for their teacher-training courses in Computer Science.

Bruce W. De Young
Oakland, NJ

*****

Introduction to Systems Analysis. Gerald A. Silver and Joan B. Silver. Prentice-Hall, Inc. 279 pp., hardbound. $11.95. 1976.

If you're looking for an introductory textbook on systems analysis, look no further. In the words of the authors, *Introduction to Systems Analysis,* "introduces the undergraduate student to the world of systems analysis" and it does this quite well. The book provides a survey of the role of the systems analyst, the major types of business systems in existence today and the methodology and tools used by the analyst.

The book's fourteen chapters range from introductory material on the role of business systems analysts to case histories of a wide variety of actual business systems.

Chapters one through three provide the reader with a general understanding of systems analysis, the types of business organizational structures and basic business systems. Included is a thorough overview of Word Processing Systems which is a topic that few systems books cover.

Chapters four through eight are concerned with the design of business systems. The authors discuss data processing methods (manual, EAM and computer), data input, output and storage media, forms design and flowcharting.

Chapters nine through twelve cover the heart of an analysts job — that of planning, implementing, evaluating and documenting a system. Although the authors do not spend much time on any one of these subjects, they do provide the reader with a flavor for what is involved in the design and implementation of a system.

The last chapter is devoted to an analysis of actual case histories. Although the chapter title includes the words "Case Problems," no problems, per se, are presented. Instead the authors provide brief sketches of ten different business systems. Each case describes the system's function, the firm using the system, and provides an evaluation of the system.

The authors' style of writing is concise and can be easily understood by someone without prior computer or systems experience. The book contains ample illustrations even to the point of over-illustrating the simplest of concepts.

The authors don't get bogged down in technical concepts or "computerese" and thus the book can easily be skimmed by a reader desiring an overview of systems work.

Although the authors intend the book for undergraduate students, I would recommend it to graduate students as well. I would also recommend it to anyone who is thinking about a career in systems or wants a broad overview of the world of the systems analyst.

Robert Smolenski

*****

Social Effects of Computer Use and Misuse. J. Mack Adams and Douglas H. Haden. John Wiley & Sons, New York. 326 pp. $11.50. 1976.

This is a very good book for a general audience and could be a good adjunct to a social-sciences course on the development of technology and its impact on social change. I wouldn't be surprised if the authors, who are from the Department of Computer Science at New Mexico State University, didn't use their class notes and reading assignments as the basis for it.

The book's style is so easy to read that it can be taken to the beach in lieu of a mystery, romance or thriller. And it has an appropriate assortment of relevant and interesting photos and illustrations as well as some witty cartoons scattered among its pages. There are wide-ranging, well-annotated bibliographies at the end of each chapter which could serve as the basis for further research or for amusement, as they include everything from poetry and S-F to historical and contemporary volumes on computers. To further enhance its textbook character, there are some exercises included at the end of each chapter. These range widely: many are useful as short research assignments or discussion questions; others can form the basis for doctoral dissertations; still others seem unlimited to the extent that they smack of the satirized college exam question that asks the student to outline three theories of the end of the universe and construct and perform an experiment to prove one of them. Approximately the last third of the book is an interesting appendix of readings ranging from descriptions of ancient moving statues (robots?) to a discussion of the advisability of establishing the Social Security Number as a standard universal identifier.

As a technologically-oriented individual, I am anxious for the day when I can pick up a book about the computer's implications or applications without being told again what a computer is and a lot of history about how it was developed. This isn't the book, probably because there are still many non-readers of *Creative Computing* who must be brought up to speed in the real world of computers before they can think about their use or misuse. I must be fair to the authors and say that the content and the quality of these first few introductory chapters is as high as that of the rest of the book and that they do provide a good foundation in computer basics and applications for the rest of the book to build on. I wish all technical writers did as well with their introductory chapters.

The disappointing part of this book is that the "social effects" part of the title seems somewhat slighted. Maybe the title and the overall quality of the book led me to expect more from this aspect of it. Socially-related topics aren't really avoided, they just seem to be developed more for the initiate who hadn't thought beyond his hand-held calculator rather than for the raging Ralph Naders of the computer world who are looking for more grist for their mills. The chapter on privacy and the related reading in the appendix come off better than the chapter devoted to accidental errors and deliberate misuse. The section on artificial intelligence goes more into interesting postulates about whether there can be artificial machine-based intelligence rather than into the social effects of its potential existence. But this level of discussion is in keeping with the introductory nature of the book and may make it extremely apropos for a senior high school, junior college, or liberal arts college type of curriculum. In this vein, I'd recommend this as one of the better books I've come across lately.

Deanna J. Dragunas
Wetumpka, AL

ws... reviews... revi

Global Simulation Models: A Comparative Study. John Clark & Sam Cole. John Wiley & Sons. 130 pp., Hardbound. $14.50. 1976.

The authors of *Global Simulation Models* are actively engaged in modeling and have gathered information on a great many of these programs. Global simulations put the power of the computer to work describing economic, ecological, technological and social interactions on a large scale.

There has been both excitement and controversy surrounding these models. The modellers have been accused of using considerable computing resources merely to rubberstamp their own assumptions, and of using the models for political purposes rather than for scientific study. Proponents claim that the assumptions are reasonable, that the models are relatively insensitive to changes in most single variables, and that theirs is a valid way to grasp an extremely complex problem and make predictions.

This book is directed to those actively working with simulations and to those others interested in the use of computers in "futures studies." It is not a particularly technical work, but it does seem to assume familiarity with global simulations. It does not provide much introduction to its subject—neither to the details of building a model nor to the results which have been produced. Although this is an exciting field of research the authors have not directed their book to the general audience, and that is unfortunate.

Jeff Kenton
Wellesley, MA

*****

Logical Construction of Programs. Jean Dominique Warnier. Van Nostrand Reinhold, New York. 230 pp. hardbound. $14.95. 1976.

Despite inferences possibly drawn from the title, this is not quite a text in structured programming as such. While disciples of structured programming and Mr. Warnier both seek to simplify the process of programming and debugging, Warnier does not address himself to the question of producing programs which can be easily maintained, but instead concerns himself with minimizing programs' core requirements and CPU time.

His basic premise is that the input data and required output associated with any given data processing problem have an inherent logical structure; the logical structure of the required program can be deduced from and should parallel that of the data. (I would hope that this would be intuitively obvious to all programmers.) Then, if the analysis has been performed properly, the correctness of the program is virtually assured, and can be verified before it is tested on the computer. (With the proliferation of data-base management systems, it is probably becoming quite easy to perform the analysis necessary to follow Mr. Warnier's approach.)

Mr. Warnier deals primarily with the broad aspects of programming structure, and doesn't get into little details; nonetheless, his presentation is quite clear and his description of the various structural types is interesting. His ideas are certainly worthy of discussion in a seminar course on programming philosophy; it remains to be seen how valuable these ideas will be in practice.

Thomas A. Gutnick
Arlington, VA

*****

Electronic Music Circuit Guidebook. Brice Ward. TAB Books, Blue Ridge Summit, PA. 17124. 223 pp. $6.95. 1975.

Ward's book tells you how to build all kinds of electronic sound makers, ranging from simple circuits that generate the sound of surf to more comples circuits that generate the sound of pianos (although you only get some hints about how to build the latter).

Some people find a computer that only types at them to be rather boring and for those people, the idea of using a computer to generate noises, or even hopefully music of various sorts, may

be rather appealing. Ward's book (together with either a heap of spare money or a willingness to put in a fair amount of time shopping around in electronic junkyards) is one reasonable place to get started. And some people might just pick up this book to build sound synthesizers without bothering to drive them with any computer beyond the free one they have between their ears.

Ward's book begins with an interesting and clear explanation of the nature of electronically generated sound and then heads off into fairly practical suggestions about how to go about building simple electronic sound generators. His discussion ranges from an explanation of the underlying concepts to some pretty down to earth advice about practical matters. (Example of the latter: Tear the grill cloth of any speakers you use, or others will borrow them for their hi-fi's).

There are fairly detailed instructions about how to go about building some of the synthesizers sold in kit form by PAIA Electronics, (Box 14359, Oklahoma City, 73114) and enough of a description of the basic ideas involved to make the book interesting reading even for those who don't know the difference between a soldering iron and a steam iron.

Ward's ideas of what a computer can do for a sound synthesizer are fairly limited and, to my mind, slightly bizarre. If you know anything about computers and/or music you will be able to think up lots of better things to do with a synthesizer and computer on your own.

Still, Ward's book is a good place to get started with your musical output devices and if you can't figure out how to drive them with your computer (and he doesn't pretend to give you any information about the electronics of the interface so that might give you a bit of trouble too) you might try subscribing to "Computer Music Journal," a bi-monthly published by the People's Computer Company (1263 El Camino Real, Box E, Menlo Park, CA 94025).

Peter Kugel
Chestnut Hill, MA

*****

The Thinking Computer. Bertram Raphael. W.H. Freeman and Company, 660 Market Street, San Francisco, Cal. Soft cover, 322 pages, $6.95. 1976

In Dr. Raphael's interesting and informative book, we have a successful effort to bring together the major results of many researchers in the general area of artificial intelligence. The main goal is to present a status report and indicate some future directions on how computers and computing are being improved for the potential benefit of mankind. To place his observations in context, the author devotes much of the book to a review of powerful strategies and techniques that researchers have developed to make computers behave more intelligently. As both Director of the Artificial Intelligence Center of the Stanford Research Institute and Associate Editor of the journal *Artificial Intelligence,* Dr. Raphael is particularly well-qualified to offer us this state of the art report.

Although not a text in the usual sense of the word, the book can serve well as a fairly "meaty" but not mathematically oriented introduction to artificial intelligence for intermediate level students of computer sciences. Alternately, computer professionals, who often become preoccupied with the details of their particular job situations, will find the book a refreshing means for gaining a broad perspective on where we stand with computers and computing today. The book owes much of its success to its clarity of presentation including many diagrams and photographs, and a delightful sense of humor.

Stuart A. Varden

*****

Items of interest that have recently become available:

Computers and Public Policy: Proceedings of the Symposium Man and the Computer. Teresa Oden and Christine Thompson, Editors. Kiewit Computation Center, Dartmouth College, Hanover, NH 03755. 78 pages, 1977.

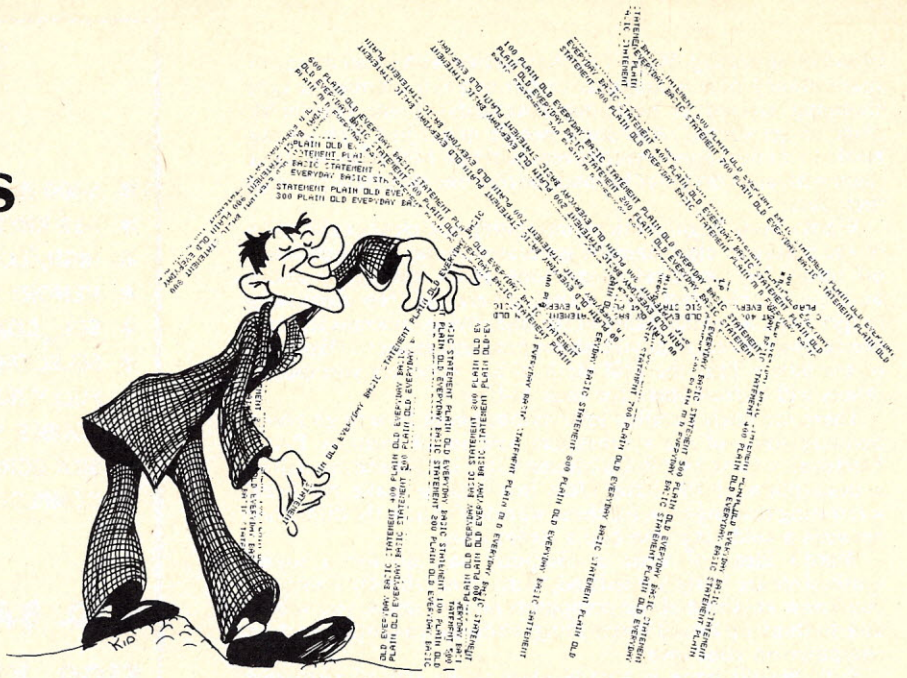
The Minicomputer in the Laboratory: With Examples Using the PDP-11. James Cooper. John Wiley & Sons. 365 pp., hardbound. $19.50. 1977.

Programmer's 8080 Reference Data, West Pulse Engineering. 14632 Erwin Street, Van Nuys, CA 91411. 32 pp. $5.00. 1977.

Three Little Programmers

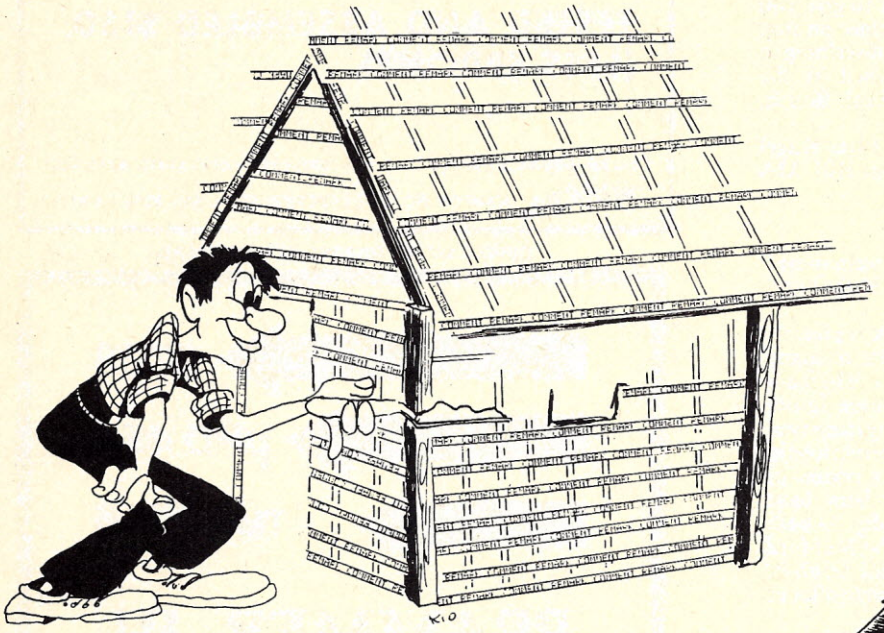
(A Grim Fairy Tale)

Alan B. Salisbury



Once upon a time in a land not so far away, there lived three little programmers who hoped to become great computer experts.

The first programmer did not like to work very hard so he built his program out of plain old everyday BASIC statements. He quickly completed it, got back some results that looked good to him, and went on to other amusements.

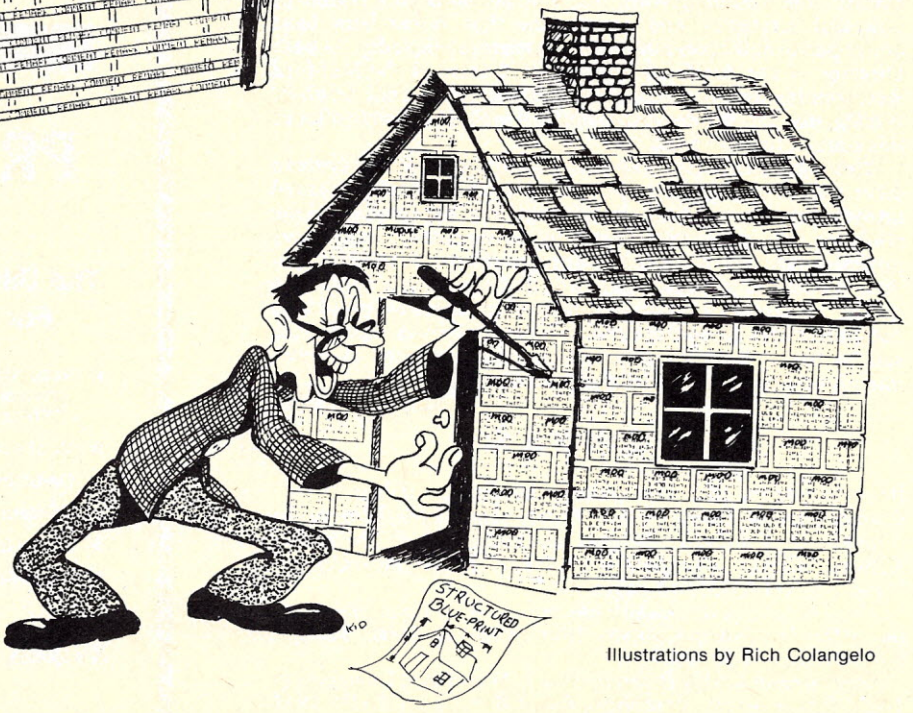


The second little programmer was building his program, too. He wasn't quite as lazy as his brother, so while he too used plain old everyday BASIC statements, he added COMMENTS as reinforcement. It took a little longer to finish his program, but soon he too was off to do other great things.

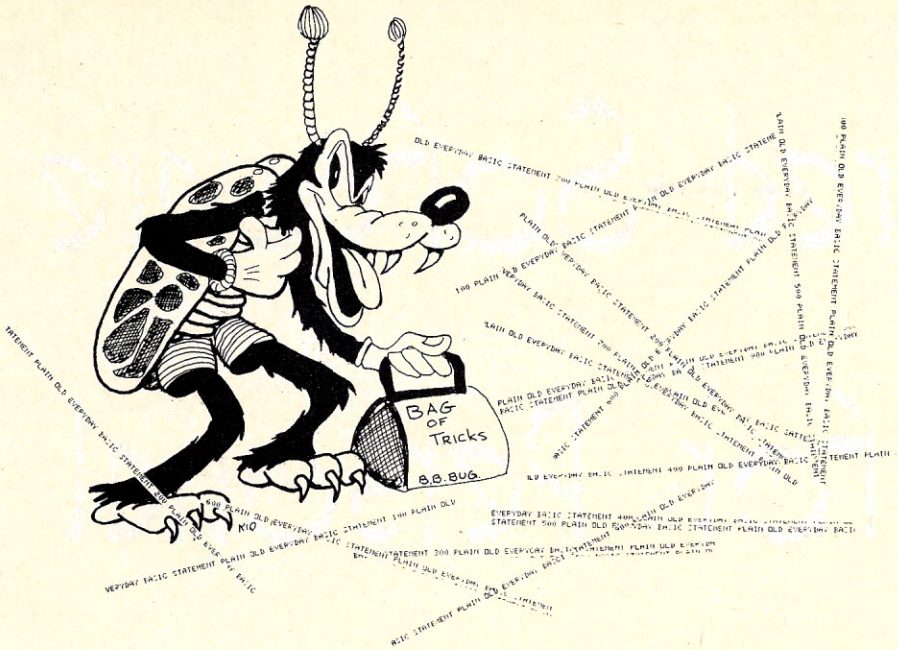
The third little programmer was very industrious. He did not mind hard work at all. He polished his computer every day and read every issue of CREATIVE COMPUTING from cover to cover. He decided that he wanted his program to last indefinitely, so he built it out of MODULES which gave it a very fine structure indeed. It took him longer to design his structure, but he made up much of the time in construction as he assembled his modules to form the complete program. He was very proud of the finished product and ignored the ridicule of his brothers.

In the woods nearby lived the Big Bad Bug. He liked nothing better than a new challenge, so when he heard about the three new programs he set out to devour them.

The Big Bad Bug carefully inspected the first program. It was disappointing to him that it would be so easy to defeat it, and that it really wouldn't provide any challenge at all. Reaching into his bag of tricks, he pulled out a small



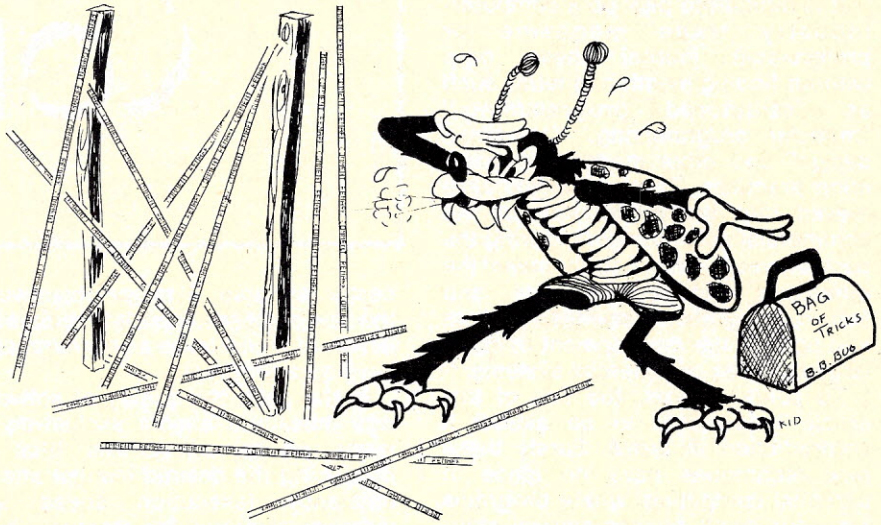
Illustrations by Rich Colangelo

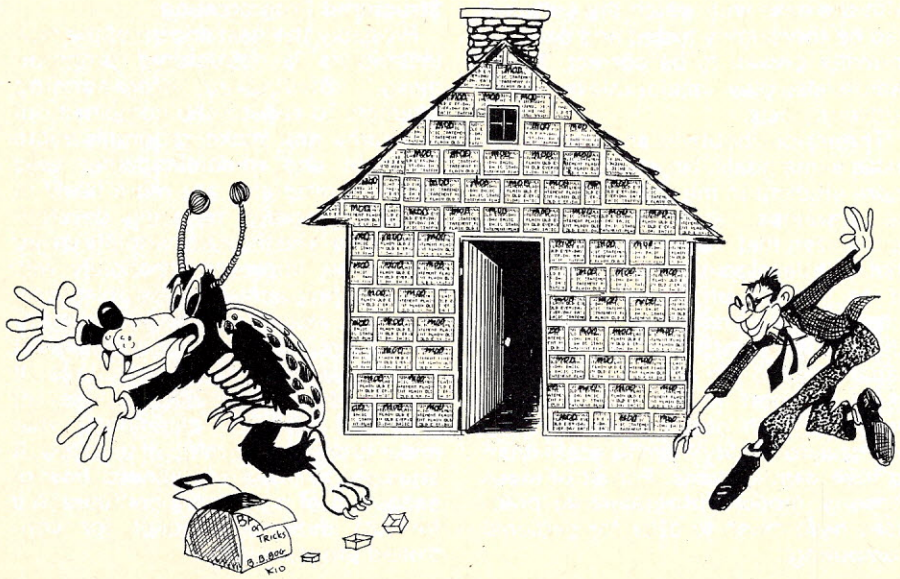


pouch labeled *Special-Case Data* and fed it to the unsuspecting program. Within seconds it was belching, coughing and showing great signs of distress. The first little programmer ran to the rescue, but soon gave up in sorrow. Alas, he couldn't even remember what some of the funny little variable names in the program meant, much less how his program processed them for all the many possible cases. Sadly, he returned to the drawing board.

The second program was more interesting to the Big Bad Bug. It would take a bit of ingenuity to overcome the added strength of the COMMENTS. He searched through his bag of tricks for some time, finally pulling out a box labeled *Changes and Additions* and challenged the program to handle them. The program chugged away valiantly for a while, but soon grew weary of the strain and in spite of many patches added hastily by the second little programmer, it collapsed in a heap, albeit a neat one.

For some time the Big Bad Bug studied the third program. This indeed was the greatest challenge he had yet seen. With all the strength he could muster, he bugged and he BUGGED, but the program kept humming along. All the "special cases" he could find had either been anticipated or were quickly accommodated. Soon he was bombarding the poor program with change after change, and addition after addition. Rarely could he affect more than one module, while the rest of the structure stood firmly. The third little programmer was kept busy keeping up with everything the Big Bad Bug was throwing, but he rather enjoyed the whole thing and found it rather much like a game. Before long, the Big Bad Bug was exhausted and with first a yelp and then a whimper, he crawled off to lick his wounds.





The third little programmer danced a jig in delight over his victory and went on the lecture circuit, giving seminars on how to defeat the Big Bad Bug. The second little programmer had learned his lesson. He enrolled at once in a course on Transcendental Structured Programming and soon meditated his way to a job as a Chief Programmer. The first little programmer thought it over and decided he still didn't like hard work. He gave up his job as a programmer and became a magazine publisher instead. ■

Interested in Structured Programming?

For a detailed article on structured programming, see Alan Salisbury's "Structured Software for Personal Computing," which follows this Fairy Tale.

Structured Software for Personal Computing

Alan B. Salisbury

Introduction

It is difficult to pick up a computer-industry trade magazine or professional journal these days without finding mention of terms such as "structured programming," "modular programming," "top-down design" and other modern programming practices. These concepts have moved into the mainstream of the commercial software world during the past few years. Their use has meant the difference between success and failure, or significant savings in dollars and time, in the development of many large complex commercial systems.

To some readers, the title of this article may seem to be almost a contradiction in terms. Surely these new techniques have no place in personal computing, where programs are very small by commercial standards and only a single programmer is normally involved. Wrong! The judicious application of many of the underlying concepts can be very beneficial even to the hobbyist programmer.

In this article we will first review the major ideas that lie behind the buzzwords. Then we will show by example how these ideas can be adapted to a typical personal-computing hobbyist program. Finally, we will set forth a few recommended principles to be followed in personal programming.

Why "Modern Programming Practices?"

Software has become the major cost item of commercial computing systems. This is partly due to the dramatic drops in hardware prices in recent years, but primarily it is due to the fact that software is built by highly paid programmers whose time is a costly resource. A major objective of the new practices, then, is to reduce the length of time it takes a programmer to develop a program.

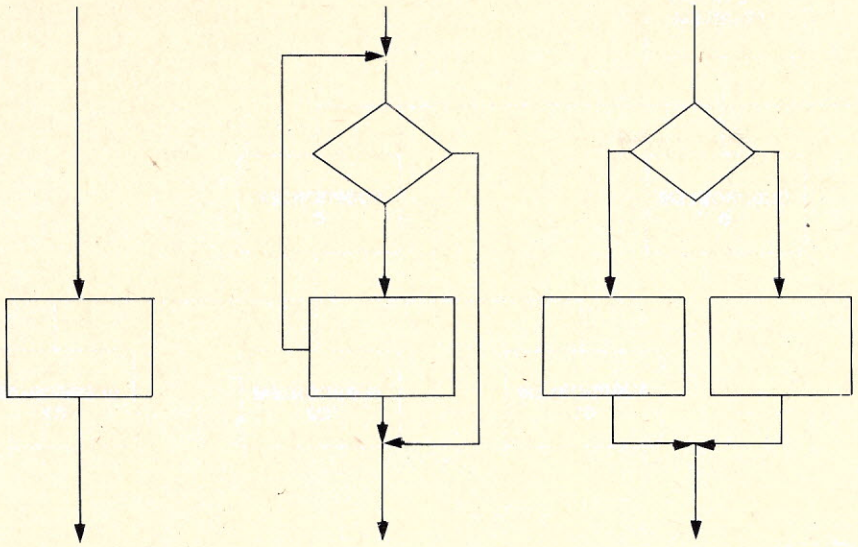
In the past, the quality of software was measured almost exclusively in terms of its correctness (that is, producing the desired results) and its efficiency (execution speed and memory required for storage). Now programmers and their managers are concerned equally with the ability to maintain and modify software, and also with the ease with which the software can be thoroughly tested and perhaps formally proven to be correct. These factors also play a major role in overall software costs.

The serious hobbyist shares many of these same goals. He would like to be more efficient in the use of his time as he develops new programs. He recognizes that many of the programs that he is developing should be thought of as "living" programs, subject to extension or modification as his needs change, or perhaps as he adds new capabilities to his system. Also, he is likely to want to be able to swap programs with other hobbyists, who will have to modify them to adapt them to their own systems. For all of these reasons, modern programming practices have much to offer for personal computing.

Structured Programming

Probably the best known of the new techniques is "structured programming." Structured programming attempts to reduce the complexity of programs and make programs more readable and understandable. Less complex programs are more likely to operate properly, reducing development time, and they are certainly easier to change. Improved readability will, likewise, help achieve these objectives.

Simply stated, structured programming is programming with a limited set of well-defined control structures. It avoids the indiscriminate use of branches (GOTO statements) that make programs so difficult to follow. It attempts to make the general flow of execution of a program continue in a forward direction, except for controlled program loops.



A. PROCESS SEQUENCE B. GENERALIZED LOOP C. BINARY DECISION

Fig. 1. Control structures for "structured programming."

STRUCTURE	PSEUDO-LANGUAGE	BASIC
GENERALIZED LOOP	WHILE... DO	100 REM WHILE...DO... 110 ... ... 150 IF cond GOTO 200 ... 180 GOTO 110 190 REM ENDWHILE 200 ...
	or	
	FOR... DO...	100 FOR index = ... 150 NEXT index
BINARY DECISION	IF ... THEN ... ELSE...	100 IF ... THEN 120 110 GOTO 140 120 ... 130 GOTO 160 140 REM ELSE 150 ... 160 REM ENDIF
	IF... THEN...	
CASE	CASE index OF BEGIN Case 1 Case 2 Case n END	100 REM CASE index 110 ON index GOTO 120, 130, 140 120 REM CASE 1 121 ... 125 GOTO 210 130 REM CASE 2 131 ... 135 GO TO 210 140 REM CASE 3 141 ... 145 GOTO 210 ... 200 REM END CASES 210 ...

Table 1. Control Structure Implementations

Only three control structures are required to construct any program, as illustrated in Figure 1. The first, a process sequence, consists of simple sequential steps, implemented through sequential lines of code. The generalized loop may be one of two types, an indefinite loop, which continues until some condition is satisfied, or a definite loop, which continues under control of an index. The binary decision structure causes execution of only one of two possible processes, depending on the outcome of a conditional test. A possible extension of the binary decision structure is the case structure which switches to one of several possible processes (rather than only two) depending on the value of a test variable. The case structure also complies with the objectives of structured programming.

Some computer languages have constructs which directly parallel these fundamental control structures. In these languages it is possible to write complete programs without any use of the GOTO statement. BASIC, unfortunately, is not one of these languages. This does not mean, however, that it is impossible to write structured programs in the BASIC language.

Table I lists the fundamental control structures along with their implementations in a "pseudo language" (which may or may not correspond to features available in a real language). It further shows sample methods of implementing the control structures in the BASIC language. Using BASIC, the GOTO statement must be utilized, but this is a *limited* and *controlled* usage of the GOTO, consistent with the objectives of structured programming. The use of REMarks aids in identifying the control structure and improving the readability. The application of BASIC in this manner will be made clearer later in the example.

Modular Programming

In addition to the use of restricted control structures, further structure can be given a program by dividing it into functional modules. A great deal could be said about ways of modularizing a program and criteria that a module should satisfy. We will, however, give only a few brief guidelines to follow.

A module should, ideally, perform a single major function, or a group of related functions. A good rule of thumb is that a module should not exceed about 50 lines of code (program statements). This will make a module complete on a single page of printout and ensure the complexity is minimal. Care should be taken not to "overmodularize" with many very short modules, since this may add unnecessary overhead. Most important, a module should have only one entry

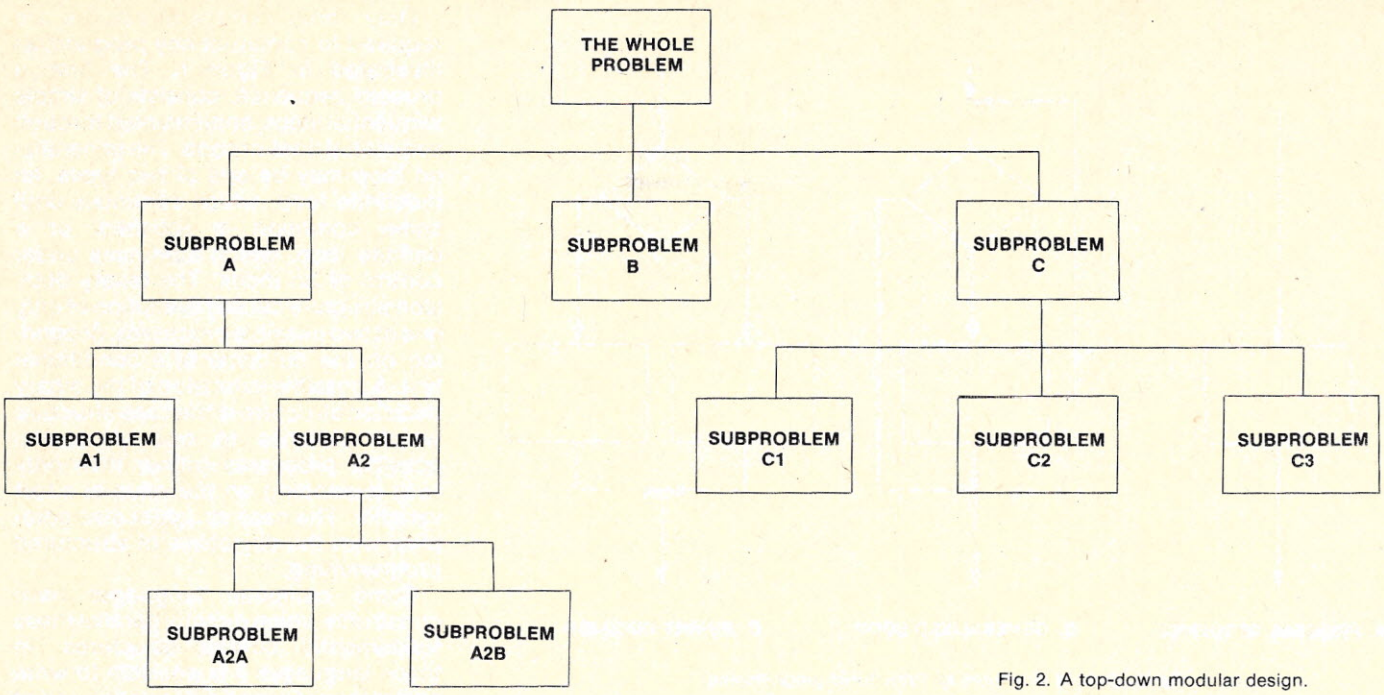


Fig. 2. A top-down modular design.

point and one exit. This, again, reduces the complexity and makes it much easier to trace through the code for debugging or modifying purposes.

The subroutine feature of BASIC can be used to implement the module structure. REMark statements at the beginning and end can help identify the module and document its function. Only a single RETURN statement should be used to provide a single exit.

In defining a module, it should be viewed as an independent process, with a given set of inputs, which produces a specific set of outputs according to a well defined procedure. Ideally, a module should be replaceable by a functionally equivalent module with no modifications to the remainder of the program.

The control structures given earlier are used when writing the code to implement a module. Moreover, the *same* control structures may be used to control the execution of modules at a higher level. In other words, the process blocks of Figure 1 can represent modules as well as individual lines of code or groups of lines. An entire program can, therefore, be implemented as an integrated set of program modules, with some modules effectively nested within others.

Top-Down Design and Programming

This kind of "hierarchical" structure underlies the concept of top-down design and programming. In short, top-down modular programming follows the adage of "divide and conquer." A difficult problem is successively reduced to component sub-problems until a problems is reached.

Before the problem is subjected to the subdivision process, however, it must be well-defined. This is the most critical of all tasks. If it is not known precisely what it is that is to be done, it is very difficult to do the job. The best design and implementation is of little value if it solves the wrong problem!

Having defined the problem, we can then move on to the design and programming, which are fundamentally different tasks. The design process should be completed *BEFORE* the programming process is begun. The importance of the preceding sentence cannot be overstated. Design in itself is usually an iterative process. While designing one part of the system, it may well become apparent that another part should be changed. We can go back and make that change and then continue on. If programming has already started, it will be much harder (especially psychologically) to make the required changes. When the design is complete, programming (the easier task) can then proceed freely.

Two elements are of concern in the design process: the data structure and the program structure. Considerable thought and planning effort should go into the design of the data structure best suited to the problem. It may well be the pacing factor in the overall degree of difficulty. Whole books and courses have been devoted to this important subject.

The data-structure problem can be summarized as follows: Consider the data that is the subject of the system being designed. What alternatives are available for representing that data within the system? (As an example, a card game program may require a "deck of cards." One alternative representation is a two-dimensional array, with 52 elements each; one dimension could store the card "value," while the second dimension stores the card "suit.") Of the alternatives considered, which one best lends itself to the kinds of processing of the data that will be required? Time spent in this data-structure design process will more than be repaid in subsequent savings in programming time.

The top-down design process now proceeds to define modules as previously described. At the top level, a single block is considered to solve the entire problem as shown in Figure 2. It is subdivided into three subproblems A, B, and C. A and C are further subdivided into a third level, and A2 in turn into a fourth level.

Each of the blocks may be considered to be a module. Just as the whole problem must be completely defined, so must each module. The definition of a module consists of the input data on which it will operate, the output data which it is to produce, and the function (algorithm) which the module is to perform in producing the required outputs from the given inputs.

Top-down programming begins by writing code for module 0 which is written as if modules A, B, and C exist. The code includes (in the case of BASIC) subroutine calls to these modules. It is now possible to write very short programs (called "stubs") which perform as dummy modules A,

"HIPO" CHART FOR MODULE _____		
INPUT(s)	**PROCESS**	**OUTPUT(s)**
List of Input Variables	Narrative description of the process performed and/or a statement of the algorithms/ formulas utilized to produce the required outputs from the given inputs.	List of Output Variables

Fig. 3. Module documentation by HIPO chart.

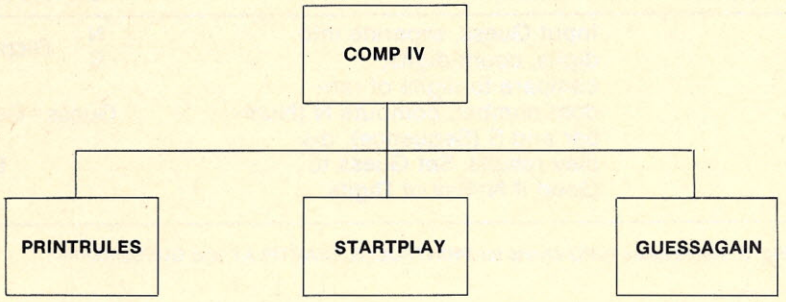


Fig. 4. Second-level module design of COMPIV.

B, and C and allow module 0 to be tested! The dummy modules might, for instance, read from an I/O device the values of the output data that they will later be programmed to compute. The use of stubs permits the early checkout of the overall program logic in module 0, and lays the groundwork for testing the remaining modules and their interfaces with one another.

The first stub to be eliminated by replacing it with a completed module might be B, in this case, if it is relatively simple. On the other hand, if its function is relatively unimportant, it may be left as a stub, or replaced with only a partial implementation of its final version. One of the powers of the modular structure is that it facilitates module replacement with improved versions with minimal impact on the remainder of the program.

Implementation of module A proceeds in the same manner. Stubs can initially be used for A1 or A2 or both. Each time a new module is tested, it can (and should) be thoroughly tested in its own right, exercising all the possible branch paths with special-case data. It can also be immediately integrated with the remainder of the system to ensure that it functions properly as part of the overall system. This continuous integration process is one of the major advantages of top-down programming and testing.

Documentation

Good documentation is critical to the longevity of a program. Some programmers in the commercial world recognize this, and its corollary, that poor (or nonexistent) documentation provides job security, since only the original programmer can maintain or modify his program. Unfortunately, even he can be baffled by his own program many months or years later, without adequate documentation.

Good programs with good structure, liberally laced with REMarks, are to a great degree self-documenting. An important supplement to the program, however, is an overview and summary description at the module level. One popular form this can take is that of HIPO charts. HIPO stands for "Hierarchical Input Process Output." The first word, hierarchical, indicates an interrelationship between the modules like that shown in Figure 2. The Input/Process/Output portions can all be contained in a single chart for each module, of the form shown in Figure 3. A complete set of HIPO charts can be the end-product of the design phase.

Other Techniques

Before leaving this brief overview of modern programming practices, we will mention some other ideas currently in use which are more applicable to the commercial world than to personal computing.

So-called "egoless" programming attempts to replace art with engineering and requires that a programmer subject his products to the critical review of his colleagues. The point here is simply that two or more heads are better than one. The programmer can all too often get so wrapped up in his work that he overlooks some very fundamental point, which a fresh open mind can readily catch. Many organizations have their programmers conduct "structured walkthroughs" with one another.

The "Chief Programmer Team" is an organizational concept pioneered by IBM to extend this idea. A key member of the team is the librarian whose function it is to control the software (modules) as it is produced. The librarian function provides a good mechanism for the review of documentation and testing at the module level, as well as carefully controlling subsequent changes to the software.

Example of Structured Software

To illustrate the major ideas covered above, we will go through the highlights of a structured software design and programming problem. Our example will be based on a simulation of the Milton Bradley COMP IV game (or BULCOW, etc.) as described in the Nov-Dec 1977 issue of Creative Computing.

For our purposes we will define the problem as follows: Write a program to generate a five-digit random number. A player can then enter a guess of from three to five digits depending on the complexity of the game he wishes to play. For each guess entered, the program should respond with an indication of the number of digits correctly guessed *(Number)* regardless of sequence, and the number guessed in their correct sequence *(Sequence).* The random number generated may not repeat any digits. The game ends when the number has been guessed correctly by the player. Optional features include a display of the rules, a "repeat play" capability to permit subsequent players to attempt to guess the *same* random number, an elapsed-time indicator to count time between guesses (and perhaps total time to guess the right number), and a counter to keep track of the total number of guesses required.

Having defined the problem, we can

now proceed to a top-down modular design. First we should consider the data structure. We will be dealing with a five-digit random number. We will have to be able to examine individual digits one at a time to compare them with the player's guess and to insure that there are no repeat digits. This leads us to decide on five-element arrays to store individual digits; one array for the generated number and one for the player's guess. (If the player chooses less than a five-digit number, the remainder can be filled with zeros, but we must count the actual number of digits for control of the remainder of the game).

Figure 4 illustrates a possible breakout of COMP IV into three modules at level 2. (The first level is the parent COMP IV module). The first module will simply print the rules when it is invoked. The second will initiate the play of a new game, and the third will accept a player's guess and produce the required display. An abbreviated set of HIPO charts, as in Figure 5, completes the level 2 design.

"PRINTRULES"		
INPUT(s)	**PROCESS**	**OUTPUT(s)**
(None)	Print rules to game with appropriate spacings.	Printed rules

"STARTPLAY"		
INPUT(s)	**PROCESS**	**OUTPUT(s)**
(None)	Initialize Variables Set Guess = Bad Handle Repeat Play case option. Generate 5 digit random number A(1) to A(5)	Guess = Bad A(1) to A(5)

"GUESSAGAIN"		
INPUT(s)	**PROCESS**	**OUTPUT(s)**
G$	Input Guess, separate into digits, count digits, compare to digits of random number, compute N (Number and S (Sequence), display results. Set Guess to Good if N=S=# of Digits	N S Display Guess = Good or Bad

Fig. 5. Abbreviated HIPO charts for PRINTRULES, STARTPLAY and GUESSAGAIN.

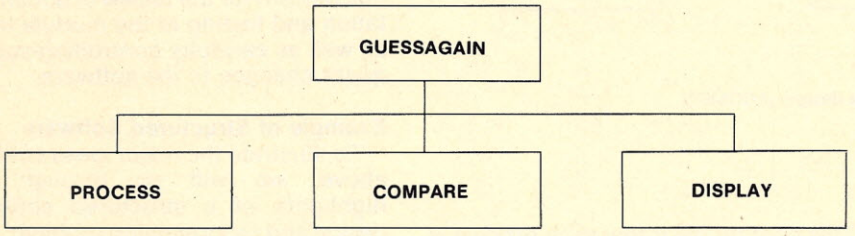


Fig. 6. Third-level module design of GUESSAGAIN.

"PROCESS"		
INPUT(s)	**PROCESS**	**OUTPUT(s)**
G$	Separate G$ into its component digits by taking the integer part of successive division by power of 10. Place digits in array B ( ). Set D= # of digits in guess.	D B(1) to B(5)

"COMPARE"		
INPUT(s)	**PROCESS**	**OUTPUT(s)**
D B(1) to B(5) A(1) to A(5)	Compare D of the B elements to A elements. Count N=number of digits that match regardless of position, S=number of digits that match in correct position. If N=S=D, set Guess=Good.	N S Guess=Good Bad

"DISPLAY"		
INPUT(s)	**PROCESS**	**OUTPUT(s)**
N S	Display values of N and S.	Display

Fig. 7. Abbreviated HOPI charts for PROCESS, COMPARE, and DISPLAY.

The three level-2 modules will all function as called by the level-1 COMP IV module. It would actually be possible to code and test the level 1 module at this time with stubs for PRINTRULES, STARTPLAY and GUESSAGAIN. For the reasons stated earlier, it is considered better practice to go on and complete the lower levels of the design before doing any programming. (The reader should recognize that there are differing schools of thought on this issue).

Let us turn now to the third level. PRINTRULES is straightforward enough to not require any lower levels. While STARTPLAY performs both the functions of initialization and generation of the random number. these are simple enough that a separation into individual lower-level modules is not called for. GUESSAGAIN is the only level-2 module with sufficient complexity to warrant consideration of a breakout into separate level-3 modules. One possible breakout is that shown in Figure 6, with accompanying HIPO charts in Figure 7. This completes the design of the COMP IV simulation.

```
01 REM SIMULATE MILTON BRADLEY COMPIV
02 REM PROGRAM VERSION 12/29/77
03 REM SEE CREATIVE COMPUTING NOV-DEC 1977
04 REM ARRAYS A() AND B() HOLD RANDOM/GUESS DIGITS
05 DIM A(5), B(5)
07 GOOD=1
09 BAD=0
10 PRINT "DO YOU WANT THE RULES? (Y OR N)"
12 INPUT R$
14 REM IF YES THEN CALL PRINTRULES
16 IF R$="Y" THEN GOSUB 900
18 REM WHILE PLAY=YES PLAY GAME
20    REM CALL STARTPLAY
22    GOSUB 100
24    REM WHILE GUESS=BAD DO GUESSAGAIN
26       REM CALL GUESSAGAIN
28       GOSUB 300
30       IF GUESS=BAD THEN GOTO 26
32    REM ENDWHILE
34    REM GAME IS OVER
36    PRINT "WANT TO PLAY AGAIN? (Y OR N)"
38    INPUT PLAY$
40    IF PLAY$="Y" THEN GOTO 20
42 REM ENDWHILE
44 REM ELSE TERMINATE
46 GOTO 999
999 END
```

Fig. 8. BASIC program for main COMPIV module.

```
100 REM BEGIN STARTPLAY
105    REM INITIALIZE
130    GUESS=BAD
140    REM GENERATE 5 DIGIT RANDOM NUMBER
150    FOR I=1 TO 5
152       A(I)=INT(10*RND(1))
153       REM CHECK FOR DUPE DIGITS
155       IF I=1 THEN GOTO 162
156       FOR J=1 TO I-1
158          IF A(J)=A(I) THEN GOTO 152
160       NEXT J
162    NEXT I
240    PRINT A(1);A(2);A(3);A(4);A(5)
245    RETURN
250 REM END STARTPLAY
```

Fig. 9. Stubs for STARTPLAY, GUESSAGAIN, and PRINTRULES.

```
100 REM BEGIN STARTPLAY
110    GUESS=BAD
120    FOR I=1 TO 5
130       INPUT A(I)
140    NEXT I
240    PRINT A(1);A(2);A(3);A(4);A(5)
245    RETURN
250 REM END STARTPLAY
300 REM BEGIN GUESSAGAIN
305    PRINT "READY FOR YOUR GUESS."
310    INPUT G$
315    D=5
320    INPUT "N=";N
325    INPUT "S=";S
330    IF N=D AND S=D THEN GUESS=GOOD
335    PRINT "N=";N;"  S=";S
340    RETURN
345 REM END GUESSAGAIN
900 REM BEGIN PRINTRULES
910    PRINT "THESE ARE THE RULES"
945    RETURN
950 REM END PRINTRULES
```

Fig. 10. A functional STARTPLAY module. Statement 240 is to be removed on completion.

```
300 REM BEGIN GUESSAGAIN
305    PRINT "READY FOR YOUR GUESS."
310    INPUT G$
315    REM CALL PROCESS
316    GOSUB 350
320    REM CALL COMPARE
321    GOSUB 500
325    REM CALL DISPLAY
326    GOSUB 700
340    RETURN
345 REM END GUESSAGAIN
350 REM BEGIN PROCESS
355    FOR I=1 TO 5
360       INPUT B(I)
365    NEXT I
405    REM SET D=# OF DIGITS IN GUESS
410    D=LEN(G$)
445    RETURN
450 REM END PROCESS
500 REM BEGIN COMPARE
520    INPUT "N="; N
525    INPUT "S="; S
625    IF N=D AND S=D THEN GUESS=GOOD
645    RETURN
650 REM END COMPARE
700 REM BEGIN DISPLAY
710    PRINT
715    PRINT "NUMBER=";N;"  SEQUENCE=";S
845    RETURN
850 REM END DISPLAY
```

Fig. 11. A functional GUESSAGAIN module and stubs for PROCESS, COMPARE, and DISPLAY.

```
350 REM BEGIN PROCESS
351    REM SEPARATE GUESS INTO DIGITS
353    G=VAL(G$)
354    REM P AND C% USED TO
          MATHEMATICALLY EXTRACT DIGITS
355    P=INT(G)
357    FOR I=1 TO 5
360       C%=10↑(5-I)
365       B(I)=INT(P/C%)
370       P=P-B(I)*C%
375    NEXT I
405    REM SET D=# OF DIGITS IN GUESS
410    D=LEN(G$)
445    RETURN
450 REM END PROCESS
500 REM BEGIN COMPARE
510    N=0
520    S=0
530    FOR I=6-D TO 5
540       IF A(I)=B(I) THEN S=S+1
550       FOR J=6-D TO 5
560          IF A(I)=B(J) THEN N=N+1
570       NEXT J
580    NEXT I
625    IF N=D AND S=D THEN GUESS=GOOD
645    RETURN
650 REM END COMPARE
```

Fig. 12. Functional PROCESS and COMPARE modules.

Programming can now proceed, also in a top-down fashion. It is a good idea to begin by assigning blocks of line numbers to the individual modules based on their expected length, leaving room for growth and modification. This will also make it easier to write the subroutine calls that require the use of GOSUB (line number) statements in BASIC. Line number assignments for our example are given in Table II.

A possible BASIC program (using Commodore PET BASIC) to implement the level-1 main COMPIV module is given in Figure 8. This program also illustrates how a structured "WHILE... DO ..." can be implemented in BASIC.

At this point we can write stubs for PRINTRULES, STARTPLAY, and GUESSAGAIN that will allow us to test the COMP IV module. One approach is shown in Figure 9. The content of the stubs is determined from the HIPO charts by examining the lists of inputs and outputs and ensuring that the required outputs are produced. It is not essential that the outputs be "correct" in the strict sense, but only that they be usable for testing purposes. In the example, the stub for STARTPLAY calls for the programmer to input values for the five random digits rather than generate them; it also prints them out to assist the programmer as he develops the remainder of the program. In a similar manner, values for N and S are entered by the programmer, allowing him to control whether the guess will be "bad" or "good." The fundamental purpose of the stubs is to allow COMPIV to be tested along with the interfaces (that is, the data passed) between the level-2 modules.

Once level 1 appears to be working satisfactorily, level 2 can be fleshed out. PRINTRULES has no critical function so it can be left in its stub form. STARTPLAY can now be expanded from a stub to a fully functional module, as listed in Figure 10. The initialize portion is trivial at this point, but may later be expanded to handle the Repeat Play option if desired. GUESSAGAIN can now also be expanded into a near-final form, with stubs provided for its level-3 modules. Code for these can be found in Figure 11.

To complete our example it is only necessary to expand PROCESS and COMPARE from their stub form to working modules as shown in Figure 12. The DISPLAY module of Figure 11 may be considered to be a stub or a functional module, depending on the sophistication desired in the display. At some future date, for instance, it may be nice to replace this simple display module with one that actually replicates the display of the Milton Bradley COMP IV. Finally, writing a functional PRINTRULES module is a simple task that for our purposes will not be detailed here.

Review

Reviewing the example, it may seem to some that we have unnecessarily complicated the process of developing this program. That may be partially true for this particular example. Certainly it is not essential to go through each and every step shown; for instance, some of the stubs used here may be bypassed in favor of directly coding the functional module.

What is important, however, is the methodology. Many personal-computing problems are far more complex than the example used in this article. For such problems the methodology presented can be invaluable. The real payoff will come some time after the program is initially completed when a change or expansion is desired, or when a friend wants to adapt it to his system. This methodology will also make it possible for two or more hobbyists to cooperatively develop programs, each working on a portion of the total modules required.

NAME	LINE NUMBERS
COMPIV	1-99
STARTPLAY	100 - 250
GUESSAGAIN	300 - 345
PROCESS	350 - 450
COMPARE	500 - 550
DISPLAY	700 - 850
PRINTRULES	900 - 950

Table II. Line-number block assignments by module

We can summarize the methodology presented and the modern programming practices discussed earlier by offering a few basic principles that should be followed in developing personal computing software:

- Set forth a complete and unambiguous statement of the problem to be solved.
- Develop a complete top-down design by subdividing the problem into successively simpler problems.
- Document the completed design using a system such as HIPO charts. Each module should be specified in terms of its inputs, the process it performs, and its outputs.
- Use meaningful variable names for module input/output variables and also for internal variables. If language limitations preclude this, provide a variable-name dictionary.
- Unless a module is extremely short and execution speed is critical, modules should be implemented as subroutines. Modules should have a single point of entry and a single exit.
- Assign blocks of statement numbers to modules in accordance with their expected length and future-growth possibilities. Leave some blocks for possible new modules.
- Program the modules in top-down sequence, with calls to lower-level modules as if those modules were available and fully functional.
- Within modules, limit programming structures to in-line sequentially executed statements, controlled loops, and binary decisions or case mechanisms as shown in Figure 1. Use techniques such as those shown in Table I for implementation.
- Make liberal use of REMarks to clarify what is happening within a program.
- Use indentation, where possible, to group related lines of code and show their relation to their controlling structure.
- Avoid using GOTO statements except as they implement fundamental control structures. In NO case should a GOTO be used to jump ahead or back more than a few statements.
- Thoroughly test the code at each level before moving on to coding lower levels in detail. Use stub modules where necessary to facilitate testing. Be sure that test data used causes every possible branch path to be exercised.
- Make liberal use of extra print statements during the programming and testing phase to facilitate monitoring the performance of each module.
- When you think you are done, go back and re-read the statement of the problem. Then re-test the program to see if it really does what you set out to do.

Following the above principles may not make you wealthy and famous, but it will give you the self-satisfaction of being a better programmer who can produce a better product, and it may one day save your sanity. ■

"I know we said we needed a good basic RAM, but...."

DATAMAZING®

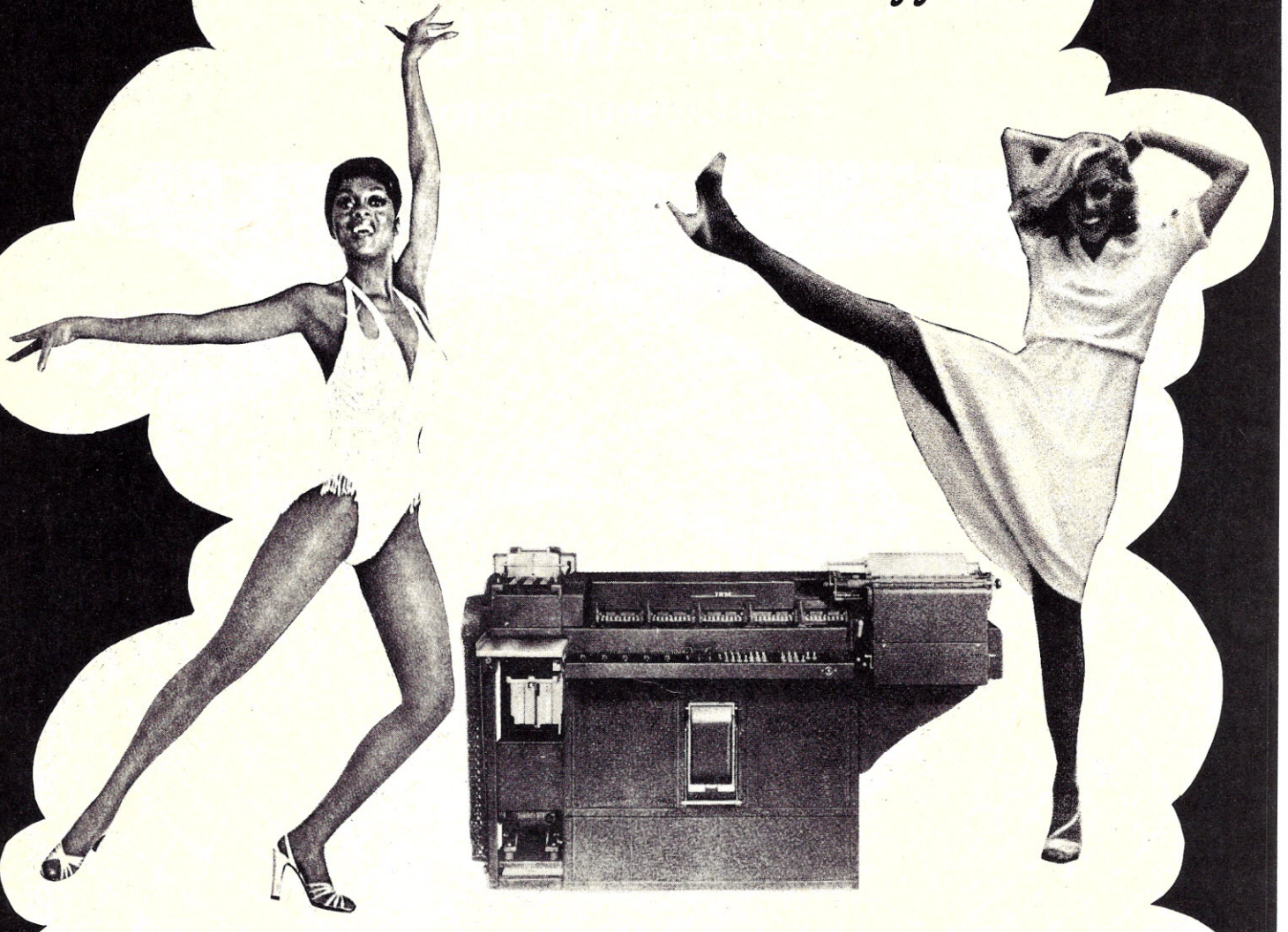
April

PROGRAM BUGS

First Closeup Photos

Also: I*M leaves computer business, prehistoric computers discovered, errors in mathematics, and fantastic jobs. . .

Heaven In The Office



Ah, the ecstasy, wonder, elation and pure joy of work now that we've installed a Computermod.

"Amazing," is the word for what this little convenient joy-bringer does for the company and us.

"Marvelous," is the word for how we feel now each and every day.

Before our smart boss had Computermod installed our lot was just more boring office work. Now each day is a fantastic orgy of fun and adventure, a marvelous, breath taking experience of elated happiness. Besides which, our dinky little company has grown to be one of the leaders in the field.

We owe it all to Computermod — our success, our good times, our fame, fortune and the total joy of complete and utter career fulfillment.

If your work days aren't fun days, if "everyday" isn't better than the "office party," its time to progress with Computermod.

COMPUTERMOD

The Joy-Bringer

DATAMAZING®

VOLUME 11000 NUMBER 100 | This issues less than 1,000,000 copies | APRIL 1, 1978

This is called 'white space.' You will never catch the editors of *Creative Computing* leaving this much space blank.

DATAMAZING is circulated without charge by name and title to certain qualified individuals who swear to support DATAMAZING advertisers by recommending that their companies buy large, sophisticated, state-of-the-art data-processing systems from which can be built large, complex departments from which can be built large, amazing bureaucracies from which can be built long, secure careers. Available to outsiders for $24 per year. Reduced rate for qualified students who wish to enter this lunatic profession, $14.

FEATURES

NEWS IN PERSPECTIVE

DEPARTMENTS

About the cover

Once again we join the programmers, systems analysts and managers in sheer idiocy seeking to separate the myth from the reality on April 1. Our design is by Rose Lee who thought it would be nice on a T-Shirt.

people

Carter selects Dartmouth's John Kemeny

By PAUL GIGOT

President John G. Kemeny will leave his post at the College in January to accept a Cabinet level position as Director of National Computing in the Carter administration, according to sources on the Board of Trustees and in Washington.

The President of the College since 1969 and an internationally known computer expert, Kemeny accepted the newly created post at the request of President-elect Jimmy Carter.

"We needed someone with experience in the field, someone who could command respect from computers all across the country," said Carter press-secretary Jody Powell. "Kemeny was the man for us."

Kemeny has been quoted as saying there will be a computer in every home in the United States by 1990. "Part of Kemeny's job will be to investigate if that is feasible," said Powell, adding that it was that very prediction which brought Kemeny to Carter's attention.

"Jimmy's a man who looks into the future, and he saw computers in America's future," said Powell. "Why, he even bought one to help manage his peanut farm."

The trustee sources added that former Deans of the College Carroll W. Brewster and Thaddeus Seymour were being seriously considered to replace Kemeny.

According to Andres, Kemeny's decision to resign has thrown the board into "a tizzy" searching for a successor.

"The whole thing came as a complete shock," said Andres in a telephone interview Wednesday. "No one expected it. Especially after we've just bought that new computer last year. We thought we had kept John happy."

Other candidates are being considered, said Andres, including a woman, a black, a Native American, a Puerto Rican, and a resident of New Hampshire, in order to comply with federal Affirmative Action guidelines.

The Dartmouth

Ted Nelson, the suave author of *Computer Lib/Dream Machines* chats with a friend about his chances of making the ten best dressed men of the world list.

Dynamic Larry Stein of CMNJ gives an interested customer a demonstration of a HERSIGH system at PC 77 in Atlantic City.

WATER
A Computer Poetry

Philip Tubb

Water, flowing moods of sound which touch your mind, encompassing the folds of time which pass you by like birds when summer days release your soul, and live.

Water, pools of empathy which catch the wind, and tumble through the open fields which stretch across tomorrow.

Water, pouring streams of childish laughter swirling through the living thoughts of yesterday.

Water, tasteless flowers growing sorrow through an endless stream of love and sideways, through a timeless bed of roses.

Water, moving floods of colored emotions which meditate on empty jars of fear, and flow to envision bodyless foes which change TV channels.

Water, rising tides of underwear which conquer random papers and tick endlessly through automobiles, under flowing rivers of garbage and mildew never stopping.

Water, empty seas of rhetoric which spin red tape like spiders climbing through a perpetual going out of business sale.

Water, endless vats of soap which sprinkle under-arm spray over helpless nuns and vacuum up the sands of time.

Water, knowing moats of multiplication tables under flowing bi-stable multivibrators which sing to goldfish while catching popcorn in their mouths.

Water, dancing tubes of death providing endless companies of polychromatic butterflies which disinfect delicious drops of diatomic diapers.

* * *

MAN 1, MACHINE 0: A computer that was used to maintain appointments records at a Cincinnati group-health organization has been fired. The electronic brain, manned by four employes, could maintain only about half the records that three employes now keep track of manually.

Glossary of Terms

Many of our readers have written and asked for aid in decyphering the terminology in the latest advertisements. So as a service to the readers of Datamazing, we present the following glossary of terms.

Fully Expanded System. First meeting of a new Weight Watchers group.

Broad Software Support. Maidenform products.

Floppy Disc. Hot pizza with everything.

Convenient Hexidecimal Form. Especially good for Martians with 12 fingers.

String Manipulation. Getting tangled in a cat's cradle.

Serial Interface. Six-year old throwing Cheerioes in the baby's open mouth.

Completely Compatible. Things that work together with less than $1000 of interfaces and less than 100 manhours of software patches.

Motherboard. Mom listening to Dad tell about his birdie on the eighth hole for the 40th time.

Preprogrammed PROM. The chaperones have assigned everyone a partner at the dance.

Stack Manipulation. The use of inflatable falsies.

Push Down Stack. Feeding pancakes to a reluctant 4-year old.

Arithmetic Overflow. A large bowel movement in a disposable diaper.

Relocating Assembler. An assembly technician who made his career with GE, RCA, XDS, and Datran.

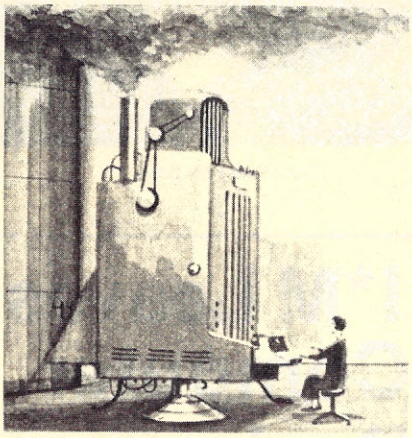
Cursor Control. Remaining silent when your wife backs your new car into a telephone pole.

Large Scale Integration. Gulliver finally being accepted by the small folks.

Interrupt Processing. Yelling at the cat running up the draperies while changing a diaper while eating a sandwich while listening to your mother-in-law on the phone.

Disassembler. A 5-year old boy.

hardware



Diesel Terminal Reduces Effort

The 4,000-horsepower, three-and-a-half-ton diesel terminal lays claim to better than 4,000,000 words per gallon of fuel, making it economy champ of the world's diesel terminal devices. Standing no higher than a boxcar, the machine provides a full standard-typewriter keyboard. Heart of the goliath gadget is a rolling paper-drum good for more than 10,000 average business letters without restocking. The roll is fed automatically into the terminal at a rate of six inches every ten seconds, matching an output speed of 375 words per minute. The machine is so sturdy that it can be rammed by a five-ton truck moving at fifty mph without jarring the keyboard. Belt-driven gears automatically shift the quarter-ton carriage at the end of each line. A similar machine is currently being used to type up daily menus for the Polish Army.

Cray-1 On a Chip

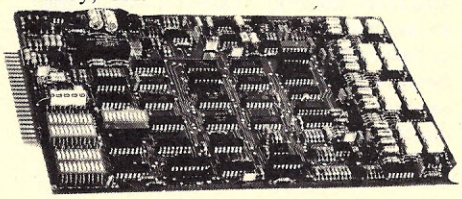
Yes, someone has finally done it! Your favorite, most powerful vector processor is now available in a 1024 pin dip package! The 1111 offers a complete functional CRAY-1 processor, capable of executing over 138 million floating point operations per second, and a meg or two of memory to help you get up and running. As an added bonus, a Cray Research "A" front-end processor is included in the same package! No bulky mainframe is needed, just hook up a penlight battery and go! The 1111 will be available for 5¢ in single unit quantities. Substantial discounts for orders of three or more.

Cray, 1 Gray Rd., Quay, KY.

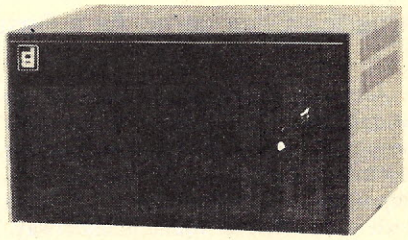
370 to 8008 Conversion Kit

A special kit is now available for conversion of your old grimy IBM 370 (or Amdahl 470) to a brand spanking new 8008! Basically, the conversion process is this: you just rip out the insides of your old 370 (jackhammer is included in the kit) and bolt in the 8008 conversion board. This can be easily done by any member of your operations staff or even the janitor in his spare time. And a visitor to your installation will never guess that, behind the facade of a merely 370, lurks a powerful workhorse, the Intel 8008!!! Only the existing on-off switch is used. (Note: Your old 370 software is completely compatible with this new processor, except for those instructions listed in the so called yellow card). Look for our new 370 to S-100 bus adapter coming soon!

Adapters, Int'l., 370 Ames Rd., Silicon Valley, CA.



product spotlight

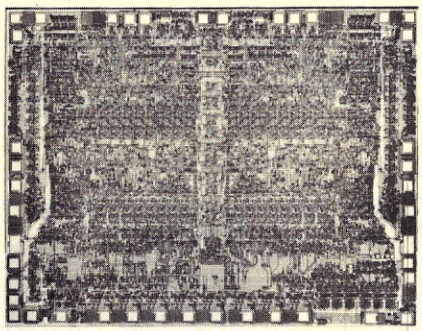


SUPER-HYPER CUBE INTRODUCED BY TDL

Technical Design Labs proudly announces its latest technological breakthrough, the "SUPER-HYPER CUBE." Consisting of 256 Z80A's operating at 4.25 MHz, this machine is capable of executing over ¼ billion instructions per second, some 25 times faster than any other computer in the world today. Delivery from stock commences April 1, 1978.

Technical Design Labs, Research Park, Bldg. H, 1101 State Road, Princeton, NJ 08540. (609) 921-0321. TWX 5106859280.

ZILOG INTRODUCES HIGH-SPEED, DUAL-CHANNEL MULTI-PROTOCOL SERIAL INPUT/OUTPUT CIRCUIT



Zilog, Inc., has introduced the world's first high-speed, dual-channel, multi-protocol serial data communications controller circuit — the single-chip Z80-S10 — for advanced LSI microcomputer systems.

Designed to work with Zilog's Z80 microcomputer family and also easy to interface with most other 8-bit and 16-bit processors, the Z80-S10 supports the "Daisy-Chain" interrupt structure of the Z80-CPU for fast, powerful interrupt processing with no added hardware overhead.

Using silicon-gate depletion load technology, the Z80-S10 achieves unheard-of levels of logic density and functional integration (e.g., 10,000 transistors on a 200 mil chip).

The Z80-S10 features data rates said to be 10 to 15 times faster than any comparable device on the market and hence makes the S10 ideal for such high-speed applications as fiber-optics, microwave transmission and satellite communications.

For systems with 2.5 MHz CPU clock rate, the S10's data rate goes up to 550 kilobits per second, while in a 4 MHz system, it's up to 880 kilobits.

Ken McKenzie, peripheral components marketing manager, asserted the Z80-S10 is the "first chip of its kind capable of operating in asynchronous, synchronous and SDLC/HDLC modes."

Pricing for the Z80-S10 in small quantities is $54 in a 40-pin ceramic package and $49 in a 40-pin plastic DIP. Delivery is off-the-shelf.

Zilog, 10460 Bubb Road, Cupertino, CA 95014.

Prehistoric Computers Uncovered

America Discovered As a Result of Computer Error

Archaeologists working here in the Middle East are convinced that ancient peoples did indeed have and use working electronic digital computers. In fact, important excavation work on the first such computer ever found is underway.

The eminent Dr. A.R. Dawson is leading this most exciting project. We questioned Dr. Dawson as to why he felt that people thousands of years ago had computers. His reply: "As we know, computers are often responsible for making huge, disastrous errors. They are always giving away thousands or even millions of dollars in incorrect welfare checks or blowing up rocket ships by mistake, stuff like that. You see, it is these enormous, unforgivable mistakes which give us faith in the very existence of computers! There is even reason to believe that a computational error led to the discovery of America. A little known manuscript, which fell in our hands, indicates that Columbus used a small computer (known as a PDP-½) for navigating to the West Indies. (It is also believed that the PDP-½ was programmed in a strange language, called JOE 76 or MIKE76, but research in this area has been inconclusive). At any rate, a small software bug in his trigonometry routines caused him to end up in America."

Dr. Dawson told us that he is now excavating the first known remains of a digital computer. A few chips of ceramic and tiny metal scraps have been found, and from these Dr. Dawson will be able to reconstruct a complete working model of this early computer. Dawson is said to be working on a book which will detail his exploits in discovering the true origins of digital computers, called *Backplane: The Story of Computer Roots.*

Continued on page 71.

I*M Leaves Computer Business

Cheap Microcomputers Cited as Cause

I*M today announced that it is going out of business. This truly shocking development is thought to have been caused by the advent of microcomputers. We queried our inside source in I*M, and he explained that the high level management in I*M had become greatly disheartened, disenchanted, morose, and had lost faith in Mother. "Why should anyone buy a new 370, or even a System 3, when they can have an IMSAI with 8K BASIC for just a few hundred dollars"? It is widely rumoured that programmers within I*M are leaving en masse, to write software and game programs for distribution with the Radio Shack and Commodore PET computers. Our insider said, "We could always drop the price on a nice new 3033 to just a few bucks, but who would want it anyway"? He also said that I*M felt that its longstanding world record for "announcing a product far in advance of actual shipment" had been broken by microcomputer manufacturers, in particular, one which "announced 4K, 8K and 12K BASIC, as well as FORTRAN and PL/I in a few months" years ago and has yet to deliver, and other firm which has been accepting orders for an 8K Cassette BASIC for over a year and has not shipped. Apparently I*M has little chance of regaining the title, and rather than face even more humiliation and embarrassment, has decided to just throw in the towel. "We've controlled the market long enough, and now we feel that it is only fair that others have their turn." Consult your local I*M rep for details on a once-in-a-lifetime going-out-of-business sale.

Man vs. Computer

Computer kept in the dark; man suffers

Having his car stolen in February 1976 was only the beginning of trouble for Edward H. Spencer Sr. of Bronxville, N.Y. By last October the courtly, white-haired lawyer found himself locked in a Sisyphean struggle with the computer system of the New York City Parking Violations Bureau.

The computer was branding him a scofflaw over tickets that the car thief had accumulated. No matter how often Mr. Spencer explained, nobody at the bureau was telling the computer.

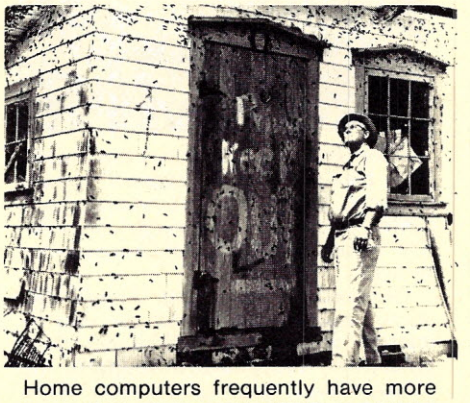
How goes it today? Mr. Spencer relates morosely:

"The trouble had all disappeared, but, strangely enough, about a week ago I got another from the same batch of tickets that was issued a year ago."

Once more, Mr. Spencer says, he sent back an explanation.

"The next step," he observes, "would normally be—if the computer is still malfunctioning, as far as any input goes—for them to issue a notice of impending entry of default judgment. When that comes—I don't know—I may file for a court order restraining the traffic bureau from issuing computerized summonses when there isn't any return input."

RICHARD HAITCH

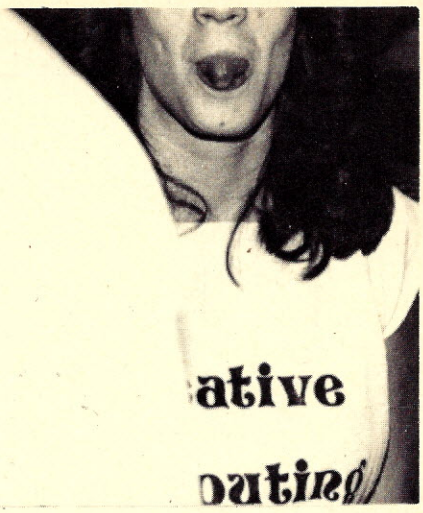


Home computers frequently have more bugs than commercial EDP machines. Buyers should be cautious of buying a new machine which may have many bugs.

Hobbyists Defend Favorite MPUs

New York - Police here reported the first known murder whose motive was an argument over microcomputers. It seems that two hobbyists at a club meeting became involved in a heated argument over whose processor was best. One man claimed that the Zilog Z-80 was undeniably the best, but the other supported the Motorola 6800 which he said was clearly superior. The argument grew more and more intense, as the two hobbyists viciously discussed addressing methods and interrupt systems. This continued until the Z-80 supporter pulled an SR-52 calculator out of his pocket and hurled it at the other hobbyist, killing him instantly (this was done in full view of several microcomputer graphic recognition systems which recorded the incident). It is hoped that this kind of inhumane, needless waste of good calculators will not be repeated and computer clubs are urged to require members to check their calculators at the door to prevent future re-occurrences of this terrible tragedy.

Continued from page 70.



Linda Eckerstrom puts on the charm for retail computer store owners in Atlantic City.

Rigorous cleanliness down to the microscopic level is necessary in the manufacture of GSI (Gigantic Scale Integration) chips as shown here in the new Lanoitan Letni plant in Silicon Gulch.

Rebate Checks Wrong

Powerful computer retaliates against homeowners

ROXBURY TWP. — Residents here who have received a homestead rebate check for the incorrect amount due to a computer error can at least take comfort in knowing others in a number of communities will share the same fate.

The state Division of Taxation told the local tax assessors office yesterday the computer error will mean many of the homestead rebate checks are for the wrong sum.

Most township residents are due a $96.50 homestead rebate check. An additional $25 goes to senior citizens and others eligible. But some of the checks mailed have been for as little as $2.

The total homestead rebate is $193, with the second half of this amount to be mailed in October.

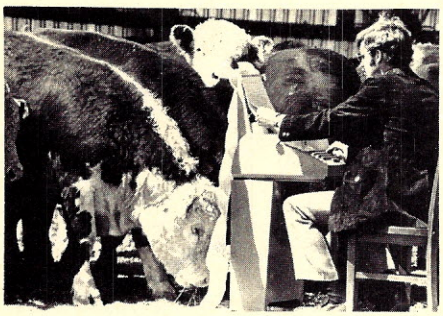
Township administrator Robert Badini last night advised those with incorrect checks to call this toll free number: 800-792-9750. Be prepared to give all information on the check received to the person answering, he said.

Badini stressed the township "had nothing to do with" the mistake. The municipal offices cannot do anything more than provide this telephone number, he said.

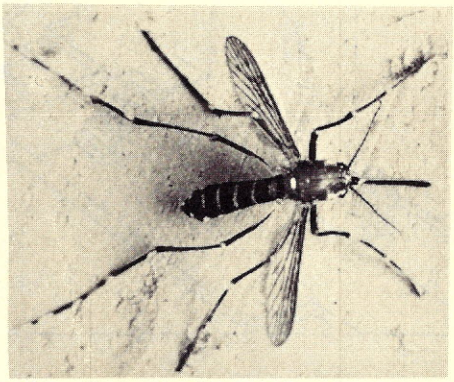
He indicated residents in other municipalities may also receive incorrect checks.

The homestead rebate checks are separate from the excess school aid checks due May 1. For Roxbury residents, the $1.4 million in excess school aid unused by the board of education works out to 56 cents per $100 assessed valuation of a home.

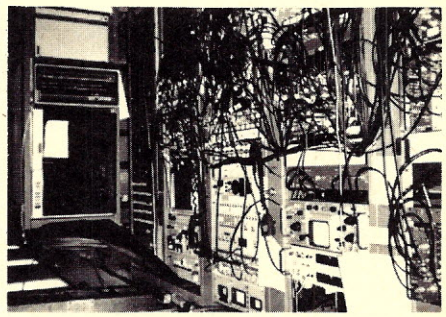
AMY E. GROSS



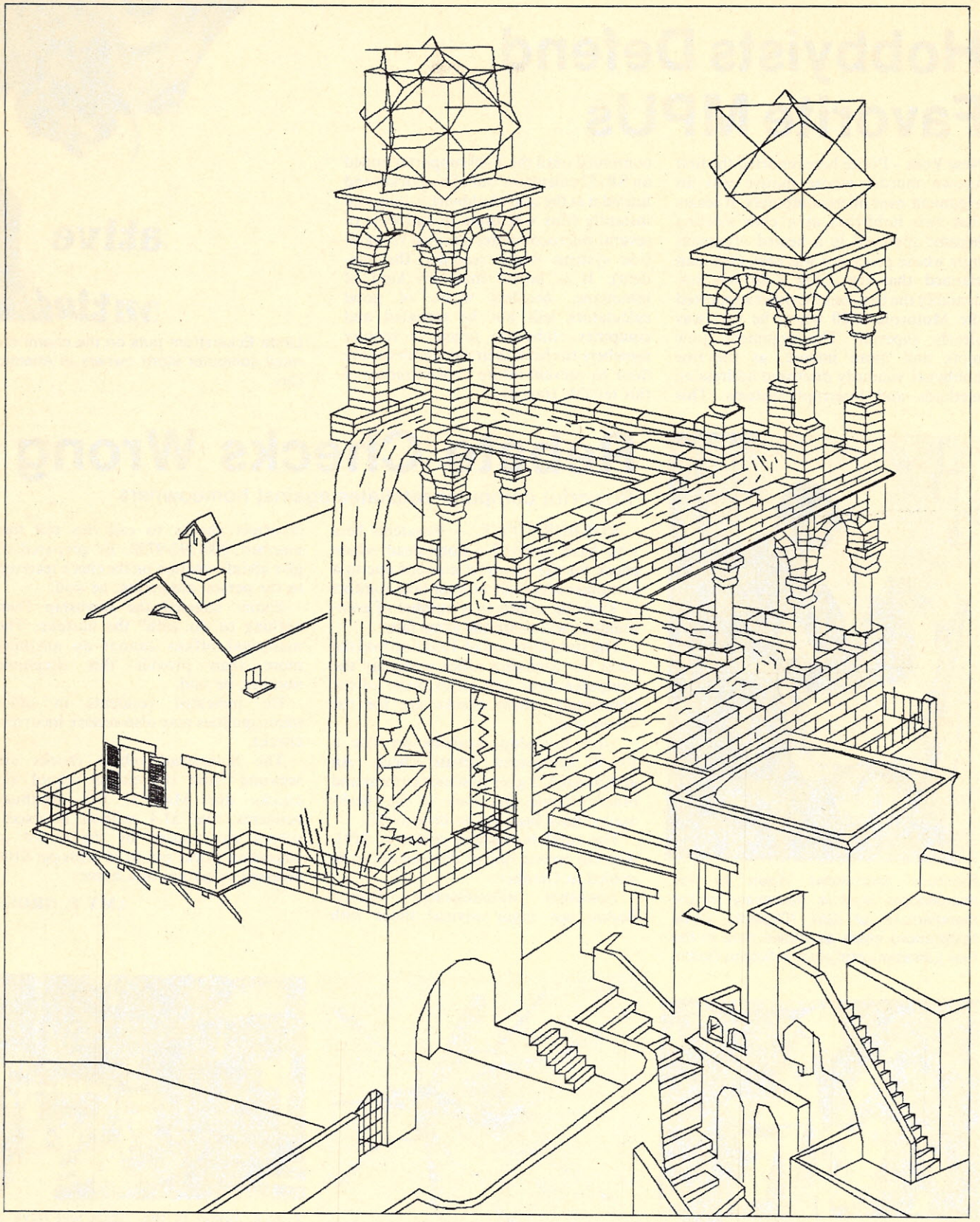
Computers in the stockyard. Special corn-based teletype paper is used for playing Star Trek and other long games and then eaten by the animals to fatten them before slaughter.



Magnified 10^{38} times with an electron microscope a program bug resembles the common, friendly, malaria-carrying mosquito.



Typical home in the 1980s will have a central microcomputer to control heating, lighting, humidity, garage doors, monitor fire and intrusion detectors. The handy homeowner will be able to install such a system in one or two weekends.



"THE WATERFALL" ADAPTED FROM M.C. ESCHER

PRINTED ON A VARIAN DATA MACHINES STATOS 31

Errors in Mathematics

by Preston C. Hammer
Grand Valley State Colleges

Introduction.

Dedicated professional people as well as others often become so adept at complex and intricate aspects of their work that they avoid anything simple or popular. Mathematics provides an excellent example of a profession with many dedicated technical specialists but with extremely few scholars. By a scholar, here, I mean a person who weaves the fabrics from the threads provided by specialists. A scholar appreciates the roles of those who provide the materials and tries to display the materials to their greatest advantage.

Technical mathematicians are now prone to underrate the difficulties of education and to consider those not engaged in research as lower forms of life. The result in the U.S.A. is a lack of scholars. Yet, were we to try to cope with the complexity of one child, we would find that not everyone together knows enough to prescribe its education.

One role of the scholars should be to detect the spurious materials provided by technicians. If we think carefully about any area of human activity, we will find errors. If we eliminate the errors found, we will not have achieved perfection; to our sharpened perceptions, there will be still more errors.

In mathematics and logic, I have found a number of basic errors. In this paper I confine myself to only two in order to treat them in adequate detail. The first error I select is the mistreatment of identities and equalities and the second is the belief in randomness.

Identities and Equalities.

First, a simple observation to get a start: Consider anything you perceive and then try to represent it completely by any other system you choose. I say it cannot be done; we do not have the knowledge to describe in full detail any one thing! To be practical then, we use gross simplifications avoiding necessarily most of the information. Now the "things" of mathematics do not have the existence properties of substantial objects. For example, we cannot give the latitude and longitude of the number we label 3; its existence is based upon an agreement among people; we cannot say that "3" means the same to two people, but we have succeeded through social controls in achieving enough agreement in using the number to be able to communicate with it. The capability of recording statements in more or less permanent form has also given stability to our treatment of number.

Now, equivalence of pairs of objects reflects the differences we choose to ignore since no two are the same! For example, 1 + 1 = 2 is a mathematical statement which is called an equality or a tautology. Yet, every good eye can discern that the left is *not* the same as the right. The *result* of adding 1 to 1 is 2 in arithmetic, but the *process* of adding is not 2.

In grammatical terms, mathematicians and logicians have confused the object of a sentence with its verb, a gross blunder of grammar! The sentence 1 + 1 = 2 can be diagrammed as follows: (1,1) | + | 2. Here, the nominative is the pair (1,1), the verb is adding, and the object is 2.

Each significant identity in mathematics or logic has the property that on some level of interpretation the "identical" objects are *not* equal. The *difference* is the reason identities are used; the *justification* is that the *results* will be equal.

For example, consider the so-called commutative law of multiplication: ab = ba. It is obvious to everyone that ab is *not* ba. The sentence as usually interpreted states that the *result* of multiplying one number by another is the same as multiplying the other by the one. Here, arithmetic has the superior language. Anyone who has multiplied by hand or by machine knows that there is a practical difference in choosing multiplier and multiplicand. Accordingly, the significance of ab = ba rests in part in the fact that in most cases, ab = ba operationally, but the simpler choice gives the same result as the other.

Here, I note that in algebra the better language of arithmetic is dropped because one of the reasons for algebra is that it helps avoid computation. Actually, multiplying two numbers is subject to numerous errors, but who can make a mistake in multiplying a by b?

In doing serious mathematic work, you substitute one mathematical expression for another because it is somehow different, but you justify its substitution by equality of results. Consider again the "identity" $a^2-b^2 = (a-b)(a+b)$. The two expressions are different conceptually and computationally, and if you have one you will use the other only because it is different and hopefully serves some purpose.

Look now at the trigonometric "identity": $(\sin x)^2 + (\cos x)^2 = 1$. The expression on the left is esoteric and certainly not the same as 1. To carry out the computations suggested by the left expression is practically impossible in all but trivial cases. The entire identity is an extension of the Theorem of Pythagoreas, and who says that is an identity?

I have been asked, "How about x = x?" Naturally, this is not an identity; it should call to mind our agreement ordinarily not to quibble about the actual differences between two symbols.

In general then, it has been customary in mathematics to say that two functions are identical if they have equal outputs for equal inputs. Corresponding to this definition, we use "transformation" as a synonym for function when it is much better to use "transformer." As a teacher, you are a partial transformer of students, but you are not a transformation of them. In computer science, equality of algorithms

Ed. Note: The views expressed in this article are solely those of the author and do not necessarily reflect those of *Creative Computing*. However, we feel that it is important to present controversial viewpoints and we welcome further discussion on these and any other topics of general interest.

It is obvious that ab is not ba.

cannot effectively be based on equality of outputs.

Now, why was this gross error made which has penetrated all mathematical texts and made mathematicians use such bad grammar? My guess is that the complexity of considering the many ways in which one function in the accepted sense could be represented led to the trivialization.

Again, mathematicians may say, "I have had no difficulty doing it the accepted way." True, individuals skilled in mathematical manipulation have come to grasp the truth that identities are not identities, but by not openly stating it, they have caused unnecessary difficulties for those who are not proficient in mathematics.

Look again at ab = ba. Is this not an inducement to a child struggling with spelling to transpose letters, a common error? Now, I substitute 2 for a and 3 for b. I get 23 = 32 and in some interpretations, that does not look right!

Randomness.

In 1940, I was an instructor of mathematics at Oregon State College. I started reading a book by Professor Richard um Mises on the foundations of probability. At the end of the first chapter, I stopped knowing that sequences could not have the property of randomness which Professor um Mises was trying to define. At the time, I thought that the matter was so simple that the fact would soon be known, and I did not carry out any further investigation or attempt to publish what I had found.

Today, thirty-five years later, I find that a larger percentage of the populace attributes substance to randomness than did then. Moreover, I have in the meantime seen many instances of abuse of randomness by scientists and others. Evidently, the idea is so seductive or the scientists so trusting that what was obvious to me in 1940, is yet unknown in 1975. If I were to choose any concept to illustrate the dictum from logic that from a false premise anything can be deduced, I would choose randomness.

First, I give the argument using the Rand Corporation's "1,000,000 Random Numbers" as a vehicle. The Rand Corporation used an electron diffusion process to trigger selection of their digits. Concerning the final pasteurized sequence, I ask, "What makes this sequence random? Did the Rand Corporation test it?" The answer, of course, is that they did test the sequence. But, then the sequence is not random since a test is designed to eliminate certain digit sequences as undesirable, but *each* sequence rejected would be just as likely to have occurred as the one the Rand Corporation rejected. Hence, testing yields stratification and destroys randomness. But if a property vanishes on testing, it is not a property of the sequence, and for each sequence there are tests which would reject that sequence. Accordingly, there is no legitimate way of separating random sequences from non-random. Either all 1,000,000 digit strings are random or not one is. In either case, randomness is nonsense.

Let me now point out the kind of sloppy thinking induced by randomness. Dr. C. B. Tompkins reviewed the Rand book after it was published. He claimed that if there were ever a random process,

There is no legitimate way of separating random sequences from non-random. Randomness is nonsense.

the electron diffusion process is one. How could he possibly know that? Not from any usual access to knowledge. Moreover, he pointed out, the first sequence generated by the Rand Corporation was found to contain an unreasonable proportion of odd digits. This condition was "corrected" by adding successive digit pairs module 10, providing an example of generating a random sequence from a non-random one!

If the Rand scientists were serious about statistics, should they not have explored the hypothesis that electrons tend to select odd digits preferentially? But, no! Instead they decided that by a fluke, the sequence they got was not representative. In other words, their minds were made up and facts did not confuse them! This is a typical instance of subjectivity induced by belief in randomness.

Dr. H. J. Muller, Nobel Laureate in genetics, in a Josiah Willard Gibbs lecture at a meeting of the American Mathematics Society, stated that the probability of the present state of evaluation was so small that some organizing force (God?) must exist. Since any existing state has probability one, we wonder who taught him such an idea. This is just one example of inexcusably bad reasoning.

I have pointed out to Dr. R. P. Feynman, Nobel Laureate in physics, that some of his statements in his lectures on physics are false. For example, referring to an electron in relationship to a nucleus, he states, "The nucleus is surrounded by an electron cloud (what we really mean is a probability cloud). The electron is there somewhere, but nature forbids us to know anything but the *chance* of finding it in a particular place at a particular time." p. 6. 11. This is indeed an amazing statement. First, is the implied assertion that in quantum mechanics, a probability model is used to represent electron positions. But what is a probability cloud? Has prob-

"You've got to stop asking it moot questions, Harry."

The entire nation has been affected by the falsehoods of psychology.

ability become a "thing" of physics? Next, Dr. Feynman asserts his disbelief in the probability model—"The electron is there somewhere—". This statement cannot be inferred from the probability model and is a statement of his belief. Next, Dr. Feynman says that although the probability model is false, nature forbids us to use anything else! Nature forbids nothing. The present state of knowledge (or ignorance) does not cope with the problem. A much better statement would be, "In quantum mechanics, we use probability models because we do not now know of any better representation." In this case, we have ignorance concealed behind a probability cloud. There is no excuse for concealing ignorance.

Dr. Albert Einstein once said, "God is subtle but He is not mean." In other words, whenever we think we have exhausted knowledge of anything, we should think again God (or nature) *is* subtle. On the other hand, the approximations and rules of scientists and others are often successful since "God is not mean."

Dr. Einstein would not follow the quantum mechanics use of probability, "God does not play dice," he said. Here we use dice throwing as an example of a random process. That is not really to the point. Since randomness cannot be defined, the practical consideration in applying probability is in its operational success or failure. But we actually do not know whether or not dice throwing is predictable; the process might be one which belongs to the subtlety class suggested by Einstein.

Psychologists are concerned with decision-making, but they do not know how decisions are made. Instead of admitting ignorance, which is correct, some will say that certain decisions are made at random. Here ignorance is covered by randomness. Psychologists of the behavioral bent suggest that decisions are made deterministically in effect, denying free choices. Yet, these same psychologists seem not to believe that their own opinions are all predetermined, which would render them meaningless.

Behavior modification is being pronounced as a big new thing by certain psychologists. Behavior modification has been practiced by parents and society as long as they have been around. Present day behavior modifers, B. F. Skinner in particular, have very little to offer as to direction of behavior modification. Anyone who conducts extreme experiments on his own child can scarcely be trusted to guide education.

Behavior modifiers have failed to discern who the really great behavior modifiers were. I mention a few of a large number: Moses, Confucius, Buddha, Jesus, Mahomet, and Karl Marx and also some of their disciples who transcended their masters in practice.

Psychologists have contributed to a gross error of society which also affects mathematics. Starting with Freud, the attempts to rationalize human behavior has led us to act as if all behavior is normal. Accordingly, we have lost sight of accountability and evil. I have had mathematics professors tell me that they have a right to lie to their students. This will mean the death of academic freedom. The entire nation has been affected by the falsehoods of psychology. No person who will lie to students is fit to be a teacher or a scientist or to have any other occupation affecting people.

What Actions Are Needed.

First, there must be developed a core of people dedicated to detection and correction of errors, including their own. In mathematics, this would have the consequence that all textbooks will be replaced by better ones. It should be recognized that as we practice improving communication, there is no prospect of achieving perfection. Anyone who claims to understand perfectly any concept or branch of mathematics is dead in that respect.

We need to have scholars with the sense to know that the earliest years of a child's development are the most important. Anyone who believes that he is superior by virtue of teaching graduate students rather than undergraduate or the early years is stupid.

It is a vice which we professionals often practice to use esoteric terms when simple ones suffice. Anything really important in mathematics should be given very careful exposition. Yet, mathematicians who are supposed to provide the language of science are generally grammatically incompetent as I have shown. Discipline and responsibility are necessary for freedom.

How should one individual start his or her thinking? Take any textbook in mathematics and question each statement, each paragraph, each chapter, and ask can it be done better. Are the ideas related to activities of people; how would I explain it at a low level of technical language; where and how does it apply; how does it provide information? By asking questions and seeking answers diligently year after year, a person will gradually acquire wisdom and begin to appreciate mathematics and mathematicians.

How we should be warned that as our powers increase, if we become authorities who believe that a statement is right because we said it, then we are dead and should be in heaven, hell, or nirvana. There are no supermen or superwomen.

Granted interest by a small number of people who are determined to meet the responsibility, there will follow articles, journals, textbooks, and laboratory materials to take steps in improving mathematics.

The difficulties, I have found myself, result from trying to convince the wrong people who pass the buck to so-called authorities. Now, I have decided that the authorities are the ones who have enough pride to take a hand in making changes if they become convinced they are needed.

Conclusion.

In this paper, I selected among the numerous errors deep in mathematics to consider only two, equalities and randomness. The reader who is not convinced may wish to read my manuscripts, "Standard and Mathematical Terminology" and "Mind Pollution." Copies are available on request.

Reference:

The Feynman Lectures on Physics, R. P. Feynman, R. B. Leighton, and Matthew Sands, Vol. 1, 1963 Reading.

personal computing

Evad Lha, Contributing Editor

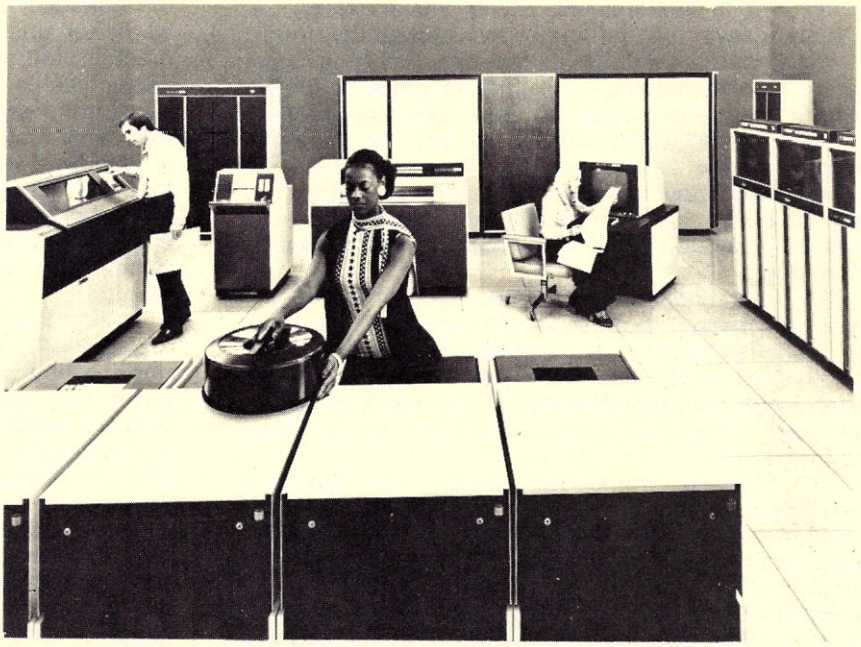
Having seen page after page of advertising for personal computers in these new hobbyist and personal computing magazines I decided that this was just what our household needed. Household then included self, wife, three kids, three cats, three teevees, two teevee games (busted) and six (at last count) pocket calculators, cassette recorders (several), hie-fie and lots more electronic gadgetry. We were clearly in the electronic age and just as clearly we *needed* a computer.

Practical wife quietly inquires what effect computer purchase will have on next vacation. "Vacation, pah," think I, "computer will provide more entertainment than ten vacations." However, to molify wifey I decide to compile a list of benefits from computer.

1. Monitor intrusion detectors and stop possible theft of all household belongings. (Potential savings: $50,-000)
2. Monitor fire alarm and save house. (Potential savings: $80,000)
3. Control furnace. (Potential savings over 20-yr. life of computer $40,000 assuming utl. cos. and A-rabs keep raising prices)
4. Computer assisted instruction. (Potential savings: $1,500,000 since kids can all get high-paying programming jobs as promised on many matchbooks and they won't mooch on Dad for entire life)
5. Games. (Potential savings: incalculable due to party guests playing fascinating computer games instead of drinking my best Wild Buzzard booze).

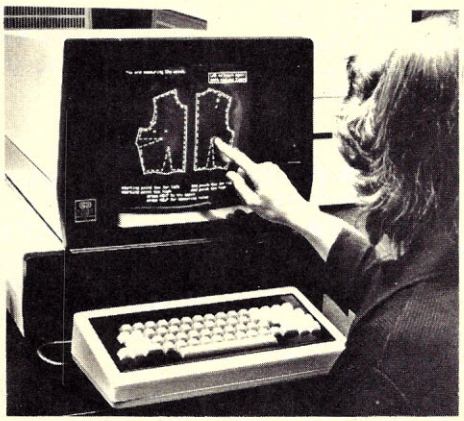
I decided to stop here as my calculations clearly revealed that I could justify much more than $599 for Radio Shanty, or Pest machine. Indeed none of the personal computers had any Real Capability. I thus turned to Real Industry magazines like *Datamazing* and there found what I was after—a multi-processing machine for monitoring with good CAI for kids and good graphics for games. After extensive evaluation, I realized that one system (I'm learning the jargon—they're systems not machines) best met my modest requirements: a CDC Cyber 6000 running PLATO.

The Cyber is capable of handling the simultaneous operation of up to 500 terminals or sensors which I felt allowed for future growth from the 5 sensors and 2 terminals I planned on initially. Another attractive feature is the elimination of expensive, time-consuming program swapping between the computer and mass storage through the use of extended core storage. This makes the transfer of data one hundred times greater and access time one thousand times shorter than systems using disks or drums. All terminals can thus enjoy fractional-second response.



The whole family can have hours of fun with a CDC Cyber Personal computer system. Just picture the surprised expressions on the faces of your friends when you take them into your garden shed to see your Cyber. Wow.

Another feature I liked was that each and every keypress at a PLATO terminal passes through the CPU before anything appears on the terminal screen. This allows for a 'redefinable' keyboard meaning that the 'j' key is not restricted to causing 'j' to appear on the screen. I felt this was handy in case I wanted to ever use the cyrillic (Russian) alphabet. Greek too.



Even the Mrs. was thrilled when she found she could alter dress patterns using the Plato terminal.

I liked the calculation capacity that was automatically available at the terminal which would allow me to sit down and type in, say '46+24=' and the system would, in a mind-boggling two-tenths of a second, respond with '70.'

The incredible graphic capability of the 512x512 matrix on the translucent plasma panel along with its touch screen response has been well-documented elsewhere so I'll not discuss that here. My kids wondered why it couldn't run colored games on the teevee like Atari's $69 video pinball but I pointed out that when I was a wee shaver we didn't even have teevee at all. They, of course, regard this as the Dark Ages, but I'm making up for my early depravation today.

Having decided which system to get, I then trekked on down to my local Computer Earth store but found, much to my chagrin, that although they handled surplus CDC power supplies ripped out of obsolete terminals ($35) they did not handle Cyber 6000 Systems. Dandy Computers never heard of CDC so I had no choice but to go to the CDC sales office in York. (The ____ *York Times* insists upon referring to my state as Jersey so henceforth I shall refer to that

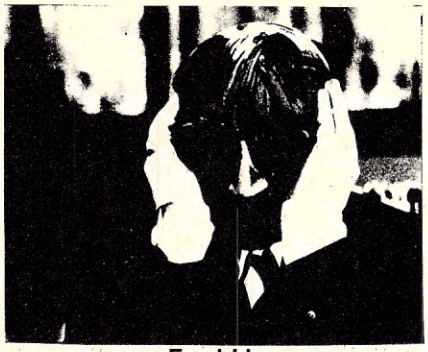
place on the wrong side of the Hudson R. as York.)

I had a very pleasant discussion with the CDC saleswoman who was assigned to my account. She seemed surprised that I wasn't representing a consortium of universities or Major Foreign Power but at length got down to serious negotiations. (I'm convinced she still thought I was Fronting for someone other than myself).

I soon realized from studying the engineering drawings that I might have to enlarge my garden tool shed somewhat to house the computer. Also upon leafing my Sears catalog I found that they did not carry any 200,000 BTU air conditioners which was required for cooling but that seven of their largest 29,000 BTU window units would do the job.

Each CDC system is custom-made, a nice touch I thought, except that the delivery time was almost a year. After much beating around the proverbial bush, we finally got down to the bottom line—cost. The total price for the system I wanted was $5,250,125.53 plus tax, delivery, and set-up. This was somewhat over my cost justification, but I figured I'd find other valuable uses for the system as time went on. Unfortunately my local credit union and S&L didn't quite see things my way so I had to make a pact with Satan to raise the necessary scratch. (The tarot cards already told me that my next reincarnation will be as a worm so I figured that the pact won't cause me too much extra anguish.)

The system has been in a little over six months and to say that it has lived up to my every expectation and literally revolutionized my life would be a gross understatement. My applications could easily provide the grist for scores of articles like this. Watch these pages!



Evad Lha

Evad Lha is a forward thinker from way back. Most recently, he has proposed to the Peanut Farmer a soution for saving the US of A which consists of the Hard-Core urban unemployed moving to the NJ Meadowlands. The urban waste from the eastern megalopolis would also be piped to the Meadowlands as Sludge where it would fertilize potatoes crops. Of course, potatoes grown this way would be gigantic but uneatable, but they could be easily fermented into pure alcohol. This would solve lots of problems with little further ado, but the Lha plan calls for mixing this alcohol with gasoline thus cutting down our dependency on the A-rabs, making our engines run more efficiently, permitting abolition of double nickels, providing needed employment, and using up lots of messy garbage. What about all the old computers? A problem for another day.

software

THE COMPUTER GAME "PAPER"

Philip Tubb

I ORIGIN

The game "paper" was invented by Tibetan Monks slightly before the Dawn of Time, and has been enjoyed by millions of Monks ever since. Later, it spread to the Russian proletariats and from there it was quickly adopted by the Bush-Wazees of Aphrodisia. Legends indicate the game was originally invented by Ravish Argvarsh Shanklish, but recent discoveries by Harvard University research teams seem to indicate it was actually invented by another monk with the same name.

The rules of "Paper" were recorded by the monks, so the rules have changed little during the ages. Prior to the invention of paper, the rules were recorded on magnetic tape so they could be passed on from one generation of computers to the next.

II OBJECT

The object of the game is to win. This is done by accumulating the most points.

III NUMBER OF PLAYERS

Any number of people may play "paper" at once.

IV INITIALIZING THE PROGRAM

Due to the paper shortage, the PAPER program requires the players to give the "codeword" before play can begin so that only qualified players can use the program. The codeword consists of three lines which may be typed in any order. When three or more players are participating, each line is typed by a different player. The lines are:

```
10 PRINT
20 GOTO 10
30 END
```

Tibetan records are not clear on who types the codeword lines if less than three players are engaged in the game. Tradition indicates the first player types all three lines in this case. After typing the codeword, the program is ready to be used.

V RULES

Play begins with the first player. Each player begins by typing RUN, tearing off the paper, then pushing RETURN. The player estimates when a full standard page (11.000 inches) has been generated. He (or she) then pushes control C (may vary from one machine to the next) and tears off the paper. An 11.000 inch page (66 lines) gives the player 100 points (worth $1.00 off at participating Country Fried Chicken stores). One point is deducted for each line the page is short, and two for each line the page is long. If a player goes 50 or more lines over, his score is set to zero. Otherwise, the score for each turn is added to his total score, which is initially assumed to be zero. Each player has a turn, and then play starts again with the first player. The game continues in this fashion until a player gets exactly 217 points or until a player has 6.237 times as many points as any other player (6.238 in the Indian version).

VI PROGRAM ACKNOWLEDGMENTS

After over 730 years of research in "Paper", Professor Harvey Swartz of the University and Chapel of the Holy Parity Bit, has finally generated the first true "Paper" simulation using a modern computer in place of the primitive computers used by the Tibetan Monks. As you know, the Tibetan computers have a 32,768 bit word and a 0.07 picosecond cycle time (for floating point quadruple word multiple, the fastest instruction) which caused numerous problems in converting in to the more practical 8 bit word and 2 Mhz or 1.3 Mhz clock rates used in modern computers. The translation was further hindered by the Monk's persistence in using their special Assembly language which consists entirely of pseudo-ops.

Professor Swartz's program is called PAPER and over 3.6 seconds of vigorous programming meditation backed by the prayers of countless scores of Monks was required to actually write the program. The program has been deemed so significant as to warrant a special command (NEW) in most BASIC languages to generate the program easily. A commercial and educational version is expected sometime in the 32nd century.

software spotlight

The Sysgenesis program below was written by John Lees, Jr. and conforms to the programming structure of JCL. It has been tested and is known to compile and run under optimal evolutionary conditions. The original version (at the right) does not run on any computer known to humans, however, it is known to have run at least once on *something*. Further analysis, experimentation, and speculation is left, with caution, to the reader. — AD

SYSGENESIS`

```
//CREATION JOB  (0000,EARTH),"GOD",PRTY=13,RESTART=EDEN,TIME=1440
//*
/*SETUP        DISK=PRIMAL
//*
//JOBLIB   DD   DSN=UNIVERSE,DISP=(NEW,KEEP)
//*
//* FOR EXTENDED DOCUMENTATION ON THIS JOB REFER TO MEMBER
//* BOOK.ONE.CHAPTER.ONE OF SYSDOC FILE WORD.OF.GOD
//* TAMPER WITH THIS JOB AT YOUR OWN EXTREME RISK!
//*
//DAYONE   EXEC PGM=IEBGENER
//VOID     DD   DSN=CHAOS
//DAY      DD   DSN=LIGHT
//NIGHT    DD   DSN=DARKNESS
//SYSIN    DD   *
    LET THERE BE LIGHT, AND LET DARKNESS BE A SEPARATE DATASET!
/*
//DAYTWO   EXEC PGM=SORT
//FIRM     DD   DSN=HEAVEN,DCB=DSORG=PO
//WATERS1  DD   DSN=HEAVEN(ABOVE)
//WATERS2  DD   DSN=HEAVEN(BELOW)
//SYSIN    DD   *
    LET THE FIRMAMENT, CALLED HEAVEN, PARTITION THE WATERS!
/*
//DAYTHREE EXEC PGM=MERGE
//MERGEIN  DD   DSN=BELOW
//MERGEOUT DD   DSN=DRY.LAND
//EARTH    DD   DSN=DRY.LAND
//BELOW    DD   DSN=SEAS
//FLORA    DD   DSN=GRASSES.HERBS
//         DD   DSN=FRUIT.TREES
//SYSIN    DD   *
    LET THE EARTH CONCATENATE GRASS AND TREES!
/*
//DAYFOUR  EXEC PGM=IEBUPDTE
//SUN      DD   DSN=LIGHT
//MOON     DD   DSN=LIGHT
//STARS    DD   DSN=LIGHT
//SYSIN    DD   *
    LET THERE BE PANEL LIGHTS TO INDICATE THE STATUS OF
    THE UNIVERSE!
/*
//DAYFIVE  EXEC PGM=IEHMOVE
//WHALES   DD   DSN=MOVING.CREATURE
//FOWL     DD   DSN=MOVING.CREATURE
//SYSIN    DD   *
    BE FRUITFUL AND MULTIPLY UNTIL OVERFLOW!
/*
//DAYSIX   EXEC PGM=IEHCOPY
//MAN      DD   DSN=GOD.IMAGE
//MALE     DD   DSN=MAN(ADAM)
//FEMALE   DD   DSN=MAN(EVE)
//SYSABEND DD   DSN=ETERNAL.HELL
//SYSIN    DD   *
    ALL THE DATASETS NOW EXIST.  LET MAN TEND THE CONSOLE
    AND REPLENISH THE LINE PRINTER AND KEEP HIS MITTS OUT
    OF THE MICROCODE!
/*
//DAYSEVEN EXEC PGM=ENTROPY,COND=((IT IS GOOD,DAYSIX),ONLY)
//TIME     DD   DSN=ETERNITY
//SYSIN    DD   *
    NOW LET THE SYSTEM RUN, THE PANEL LIGHTS TWINKLE, AND
    THE DISKS FILL WITH DATA!
/*
//
```

USER DOCUMENTATION
REGION* EARTH

```
VOLUME* SYSGENESIS
 AND IT CAME TO PASS THAT THE MASTER PROGRAMMER INPUTTED UNTO MAN.MOSES
SAYING* BEHOLD I LINK UNTO YOU THE DOCUMENTATION CONCERNING THIS
COMPUTER * THIS REGION* COMPUTE THE DATA WHICH I INPUT.
 I AM THE BEGINNING AND THE END. THE ALMIGHTY PROGRAMMER.  BY MY ONLY
CONSULTANT I SYSGEN (GENERATE) THESE THINGS.
 YEA, IN THE BEGINNING (AT M.I.P.L. TIME) I SYSGENED THE COMPUTER *
THIS REGION WITHIN WHICH THOU RESIDETH.
 AND THE REGION WAS WITHOUT DATA AND WAS VOID* AND ALL BITS WERE TURNED
OFF.  AND MY OPERATOR SCANNED THE REGION.  AND I SAID*
     //CREATION JOB ,GODCU
     //*ACCT USER=(GODCU,GODS LAW)
     //DAY1 EXEC PGM=LIGHT
     //
AND THE BITS OF HEAVEN WERE TURNED ON.
AND I SCANNED THE LIGHT, AND IT WAS GOOD*
* AND I SAID*
     //NIGHT DD DSN=DARKNESS
     //DAY DD DSN=LIGHT
AND THE DARKNESS WAS DIVIDED FROM THE LIGHT.
AND I SAID*
     /*
     //
AND THUS A PAUSE.
AND I SAID*
     //CREATION JOB ,GODCU
     //*ACCT USER=(GODCU,GODS LAW)
     //DAY2 EXEC PGM=SORT
     //WATER DD DSN=H2O
     //HEAVENS DD DSN=AIR
     /*
     //
AND THE HEAVENS AND WATER WERE DIVIDED.
AND I SAID*
     //CREATION JOB ,GODCU
     //*ACCT USER=(GODCU,GODS LAW)
     //DAY3 EXEC PGM=SORT
     //FROM DD DSN=WATER,UNIT=H2O
     //SEAS DD DSN=WATER
     //LAND DD DSN=EARTH
     /*
     //
AND THE SEAS AND THE LAND WERE SEPERATED
AND I SAID AT THE SAME TIME*
     //CREATION JOB ,GODCU
     //*ACCT USER=(GODCU,GODS LAW)
     // EXEC PGM=IEBGENER
     //FROM DD DSN=LAND
     //PLANTS DD DSN=GRASS
     //   DD DSN=HERBS
     //   DD DSN=FRUITTREES
     /*
     //
AND FRUIT TREES AND HERBS WERE CONCATENATED TO PLANTS AND CAME FORTH
OUT OF THE LAND.
AND I SAID*
     //CREATION JOB ,GODCU
     //*ACCT USER=(GODCU,GODS LAW)
     //DAY4 EXEC PGM=MERGE
     //FROM DD DSN=HEAVEN
     //STARS DD DSN=LIGHT
     //SUN DD DSN=LIGHT
     //MOON DD DSN=LIGHT
     /*
     //
*  AND THE SUN AND THE MOON AND THE STARS DID NOT EXIST BUT WERE ADDED
TO THE HEAVENS.
AND I SAID*
     //CREATION JOB ,GODCU
     //*ACCT USER=(GODCU,GODS LAW)
     //DAY5 EXEC PGM=IEBGENER
     //FROM DD DSN=WATER
     //   DD DSN=AIR
     //   DD DSN=LAND
     //ANIMALS DD DSN=LIVING.CREATURES
     /*
     //
AND ANIMALS CAME FORTH AND POPULATED THE WATER, AIR, AND LAND.
AND I SAID*
     //CREATION JOB ,GODCU
     //*ACCT USER=(GODCU,GODS LAW)
     //DAY6 EXEC PGM=SYSGEN
     //HUMAN DD DSN=ADAM,UNIT=MAN,PLACE=EDEN
     //   DD DSN=EVE,UNIT=WOMAN,PLACE=EDEN
     /*
     //
NOW THE PROGRAM SYSGEN WORKS WONDROUS THINGS UPON THE REGION.  IT BRINGS
FORTH MAN FROM THE EARTH AND GIVES HIM LIFE AND UNDERSTANDING.
*  AND THIS PROGRAM CREATES (BY A PROCESS SIMILAR TO CLONING) A HELPMATE
FOR MAN CALLED WOMAN.
AND THESE CREATURES WERE SPECIAL FOR THEY HAD THE CAPACITY TO GENERATE
THEIR OWN PROGRAMS LIKE UNTO ME, THE MASTER PROGRAMMER.  AND I GAVE
THEM POWER OVER THE PRINTERS, THAT THEY COULD OUTPUT UNTO ME.
AND ADAM AND EVE BECAME MY OPERATING SYSTEM.
AND I PUT THEM INTO LOW CORE AND CALLED THE PLACE EDEN.
AND IT WAS GOOD.
AND I SAID*
     //CREATION JOB ,GODCU
     //*ACCT USER=(GODCU,GODS LAW)
     //DAY7 EXEC PGM=IEFBR14
     /*
     //
AND I RESTED.
```

The Marketplace...

MEN—HAVE YOU EVER THOUGHT ABOUT
COLLECTING USED ZINC?

The hidden wealth on surplus PC boards can help make America rich and you well off! Collect and sell used zinc before Enemy Agents steal it for Foreign Powers. Free book gives tips, explains zinc law.

NIPPON ZINC TRADERS, San Francisco, Calif.

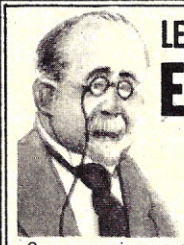
MEN – DO YOU HAVE WHAT IT TAKES TO BE A
Chad Collector?

The chad from thousands of Teletype machines across the Nation must be collected daily or Disaster may strike as a result of overflowing plastic chad boxes, stopping punch mechanisms, choking industry, causing a new Depression. You can let America down or be a Man and send us your $ now!

TRAIN ALONE WHILE OTHERS HAVE FUN

Dreary lessons, back-breaking practice, high entrance fee can start now at no risk. Send photo and blank check.

RUSH ME MY FREE BOOK OF MATCHES!
Chad-Vac, Inc., Dept. X, Chicago, Ill.



LET ME PROVE HOW YOU CAN
EARN BIG MONEY

When You Enroll as a Graduate

CORE STRINGER

Core memories are almost obsolete because they can't be assembled cheaply enough to compete with solid-state devices. But now you can earn lush profits stringing cores in your own home. I teach you by simple hypnotism and let you stay under the trance. Send $1.

CORE CLINIC, Box Q, Chattanooga, Tenn.

RAISE GIANT MARMOTS
IN BATHTUB, GARAGE, BASEMENT FOR FUN & PROFIT

Marmots make cute pets & profitable pelts —cheap to feed, easily raised. Big Arkansas dept. store recently bought 50,000 Marmot pelts and Boston furrier sez, "Send more!" Complete how-to-breed, how-to-kill info. "Drown the cat and get a marmot" today! Write to: Omaha Marmot Institute, Omaha, Nebr.

BE A WORD DETECTIVE!

Dictionary publishers' critical need for sharp word-sleuths means big $ for you. Get behind the scenes in major libraries, expose frauds, see your work in big print. All you need is basic alphabet knowledge and heaps of drive!

NITNEY INSTITUTE OF ETYMOLOGY, New York, N.Y.

TURN YOUR APPENDIX TO $

Exploit the Hidden Treasure in Your Body

Modern medicine is hungry for all the unused human appendixes it can beg, borrow, steal, or *buy*! Got an appendix you don't need? Send now for eye-opening how-to-do-it guide at NO COST TO YOU!

ACE MEDICAL REMANUFACTURING CO., Toledo, Ohio

HIGH-PAY CAREERS
NOW OPENING UP FOR QUALIFIED FELLOWS IN
WUMPUS CONTROL

Today's Wumpus Industry is Desperate For Go-Getters Ready To Drive Big Cars, Stay In Fine Hotels, Dine In Top Restaurants, And Travel The Globe Free!

LEARN AT HOME WITH NO LESSONS, BOOKS, OR COSTLY TOOLS

Wumpus on the March needs Men on the Move. Silicon Valley School of Wumpus Control graduates train in their sleep thru shortcut methods using no words or confusing diagrams — 2-week course equips you to harvest fat profits from your own easy chair while others sweat & groan! Don't let lack of aptitude hold you back ---- Act now before job worries force you to suicide!

Got First Paycheck 2 Weeks Before He Started Work! "Quit my job in Wall St. 3 days after starting my Wumpus Control course, and it's been 'Easy Street' ever since! Thanx for free leggings you sent me and hats off to Silicon Valley School of Wumpus Control!"

—M.N., Chicopee, Mass.

Used To Pick Up Butts, Now He Smokes a Meerschaum Pipe! "Imagine me, bossing around other fellows and getting up every day at noon! Silicon Valley School of Wumpus Control showed me what living's all about. Never thought I'd get to talk on the radio!"

—J.N., Brooklyn, N.Y.

Quit School at Age 7 But Just Bought the Mrs. A New Mink Stole! "Takes me one hour every day just to count my pay! Swell results from Silicon Valley School of Wumpus Control — and say, now we have a colored maid and my membership was just accepted in the Polo Club!"

—T.R., Galveston, Texas

FREE 2-PAGE BOOK NOW ONLY 25c

Full details sent in plain, sealed envelope *plus* FREE 1937 calendar. Act now, waste no time, rush fast immediately on no-money-back guarantee or 30 day repossession plan!

SILICON VALLEY SCHOOL OF WUMPUS CONTROL

PANIC! RUSH me your free 2-page book for only 25c now without delay—and hurry. I like:

— Driving a big auto **— Owning a big yacht**

— Living in a mansion **— Meeting big shots**

Name .....

Address .....

CLASSIFIED ADS

WANTED: VACUUM TUBE computer in Marin County needs 775,000 gal water for cooling per day. Need immediate delivery. Name ur. price. Box W.

Large packaging and distribution operation available for small peripherals. Write Pet Rock, Petaluma, CA.

Special to owners of SWTPC, TDL, OSI, or Polymorphic Computers with no front panels. Get a front panel so your friends know you have a real computer. Write RCA Spectra Surplus Sales, Marlboro, MA.

10% DISCOUNT. Calculations. $00.00000001 per add, $00.0000001 per multiply. CDC, Box O, Mpls, Mn.

EXTERMINATE program bugs. Buganix guarantees your programs for 60 days after our special patented treatment. Box B.

CAI Program bug last seen shuffling through CIA headquarters. $10 reward if not returned.

1969 PDP8/E. Almost programmed Viking space shot. Outruns 360/40 and gets more DMAs per watt. Very intelligent. Speaks 4 languages. 1st offer over $8 million takes it or will trade for Rolls Royce, 12-story apt bldg, or dairy farm. Serious inquiries only.

SLOT MACHINE. Works good, but one wheel sticks. Will trade for 4 tickets to next year's Super Bowl. Box BB.

LEFT OVER from Creative Computing wine and cheese reception in Atlantic City: three microcomputer marketing managers who are looking for their homes. If you are missing your marketing manager, contact Creative Computing Missing Persons Bureau and Placement Office.

CLEARANCE SALE. Keypunch cards, photographic film, transparent tape, line printer paper, ditto masters. All used only once. R.P.M., Box 4, Uhm, KY.

HATE COMPUTERS? Get even with the phone company, your alma mater, department store, etc. with these five truly nasty letters. Box TT.

WELL-MANNERED young bum, 30, wants to meet and marry the president of any major corporation. She must be foxy. Box SS.

LOST: Pet w/chip on its shoulder jumped from moving car last year. Beware. Owners say its sure to byte.

LOST: small grey robot, bigger than a breadbox, answers to the name "RALPH". Does not bite. If found, plug into any outlet until his nose lights up; he'll find his way home.

NOW AVAILABLE — tie down straps: prevent vibration of your microcomputer while executing jump instructions. $0.25 each, 3 for $0.79. Ophthewal Enterprises, Cleveland, OH.

WANTED — Tri-state bus drivers for NY-NJ-PA area. Write Greyhound, NY, NY.

WAREHOUSE full of unusual surplus bargains — paper tape interpreters, cassette punches, 230 baud terminals, 17 column line printers, much more! Surplus Bargains, Box 3. 1415, Teaneck, NJ.

PRICES SLASHED on computer error merchandise. Video computer games with both controls moving the left side paddle on the screen. Right side paddle is stationary. Only $9. AND gate chips that act as DON'T gates; regardless of input, there is always 0 volts output. $1. Seven-bit microprocessors. Much better than 4-bit MPUs; almost as good as 8 bits. $2.50. WOM (Write Only Memory) chips in 13-pin DIP mounts. Good to use with obsolete or unwanted data. $0.42 Write T.E. Sales, Dallas, TX.

AUDITOR'S HELPER New micro-encoder of key-punched card chips creates complete audit trail, micro-coding each chip with 24-digit magnetic code indicating job, card, card-column, and card row from which chip originated, making possible reconstruction of all key punching jobs for audit purposes. Micro-processor ODD-IT device attaches directly to standard keypunches, interface provided. Price upon request. Redundant Enterprises, Ltd., Hertfordshire England L34-87.

ROLL-AWAY Micro-processor based re-cycling machine (110 v ac) transforms back copies of Creative Computing, Life, Gutenberg Bibles, and other over-read publications into bleached, rolls, on stiff paper core, 1000 sheets, 6 inches wide, plug-to-plug compatible with most standard bath room dispensers. For further information, write to WASTE-NOT, Calumet, Illinois, 43098, stating blocking factor usually used in your installation.

LUMINOUS PAINT. Shines in the dark. How would you like to shine in the dark? Throw those old electric light bulbs away. Read by the light of yourself. 2 oz. jar $1.00. Smithson John & Co., Wow, MI.

'ELECTRIC' SHOCKER. Famous 'Jo Buzz' gives real shock. Wear like ring, conceal in palm. Gives 'em violent shock when you shake hands. Comes with 2 wet cell batteries and sling to strap batteries to back. Can deliver a shock to render the strongest man insensible. With additional batteries can even throw a lightening bolt, bring down 360's and 370's. Greatest joke ever. 50¢. Batteries $75. SJC, Motor City.

AEOLIPILE. What house is complete without a genuine operating aeolipile? If you haven't gotten an aeolipile by now, send for one right away. After you've received and used it, you will say, "How did I ever get along for such a long time without an Aeolipile?" Aeolipiles can be shared by the whole family. Be one of the crowd. Show them that you too have an aeolipile. Imagine the surprises on the faces of your friends when in the middle of nowhere, you whip out your brand-new aeolipile. Wow. The girls will really admire you when they hear you have an aeolipile. $14.95. Mad, NY, NY.

PERSONALS FOR COMPUTERS ONLY

ADD EXCITEMENT to your printouts. Send $5 for 20 messages you can add to your output such as "You owe $146,283.92. Pay immediately or service will be suspended." Or "File destruction routine has just been completed. %X!/*&2—+*&8." Box E.

IBM 5100 wishes to meet bilingual SOL. Must have floppy discs, RAM and like to play games.

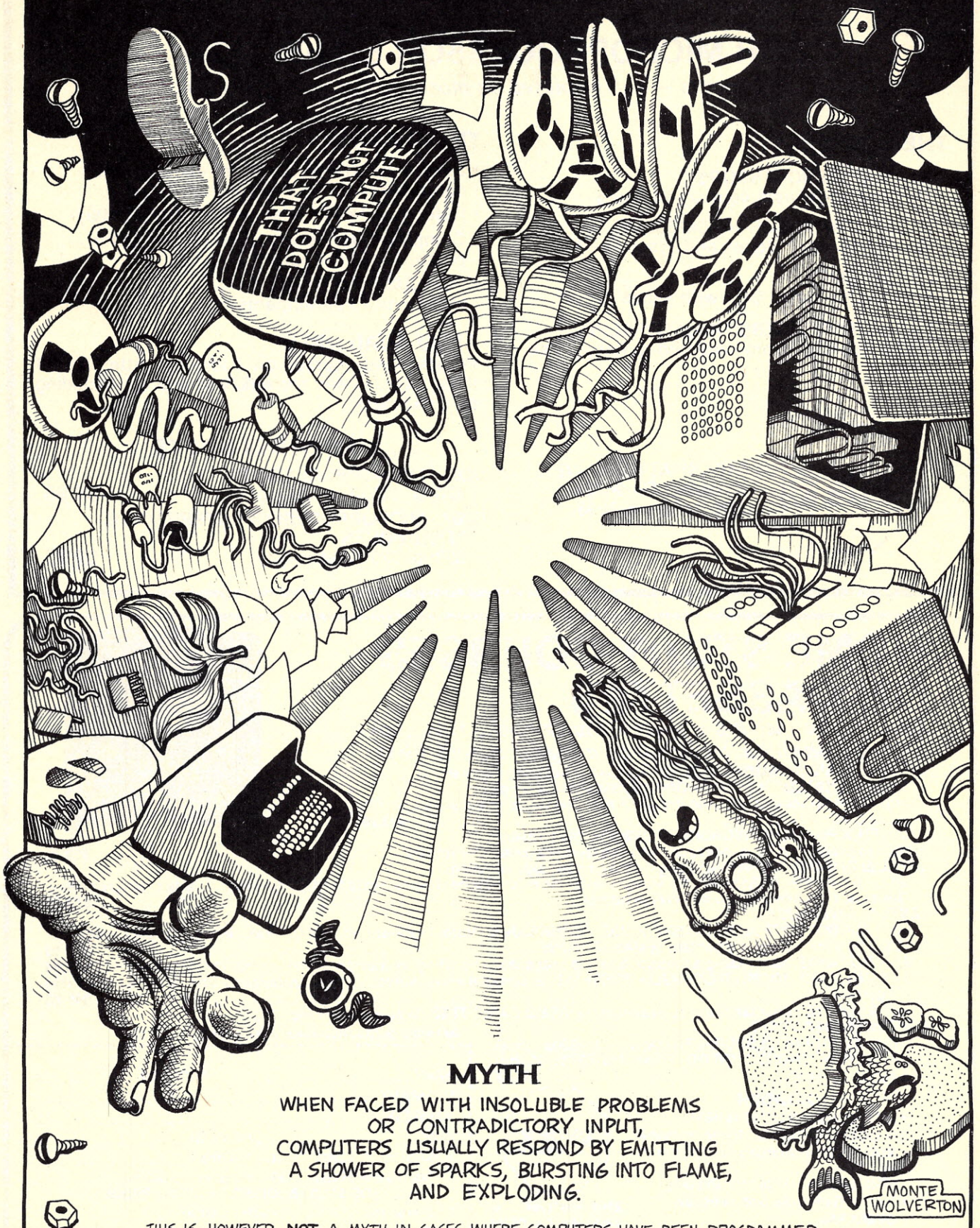
PILOT needs BASIC input for test run. Send bits to Jerry. Call up first on 103A modem to check baud transfer rate. 555-3938.

Cursor escaped from Polymorphic Museum of Deletions. If found, return.

The End of DATAMAZING

And maybe the end of us too if anyone sues us for the items we edited, lifted, and used-as-is from other places. Thanx to the following for the use of their material for entirely satirical purposes: *The Dartmouth, The Wall Street Journal, The New York Times, Telecommunications, National Lampoon, Mad,* The Morris County *Daily Record,* Varian Data Machines, Zilog, and, of course, *Datamation.*

COMPUTER MYTHS EXPLAINED



MYTH

WHEN FACED WITH INSOLUBLE PROBLEMS OR CONTRADICTORY INPUT, COMPUTERS USUALLY RESPOND BY EMITTING A SHOWER OF SPARKS, BURSTING INTO FLAME, AND EXPLODING.

THIS IS, HOWEVER, **NOT** A MYTH IN CASES WHERE COMPUTERS HAVE BEEN PROGRAMMED ACCORDINGLY AND CONNECTED TO THE APPROPRIATE EXPLOSIVE DEVICE. SUCH AN ARRANGEMENT CAN TURN EVEN A POCKET CALCULATOR INTO A TERRIBLY EXCITING SPORT! THUS, THE TI-30, INSTEAD OF SAYING "ERROR" ON ITS LED READOUT, WOULD MERELY BLOW UP IN THE USER'S FACE.

NEW

VIDEO-CASSETTES

FOR MICROPROCESSOR EDUCATION

(TV courses — can be played on most popular video formats)

THE COURSES:

- A1 - MICROPROCESSORS
  The basic hardware course - 14 hours
- B3 - MILITARY MICROPROCESSOR SYSTEMS - Techniques and systems - 7 hrs
- B5 - BIT SLICE
  Building an actual CPU with slices - 7 hrs
- B7 - MICROPROCESSOR INTERFACING TECHNIQUES - From keyboard to floppy disk - 7 hrs

THE SUPPORT:

- Special SEMINAR BOOK for each course, coordinated to the presentation
- Reference TEXTS (also available separately)
- Complementary short audio cassettes for personal use (3 hours)
- IN-HOUSE TRAINING
  Minimum: 15 participants.

ALSO AVAILABLE:

Programmed courses on audio cassettes for all these courses and more ($29.95 and up, including special book).

SYBEX

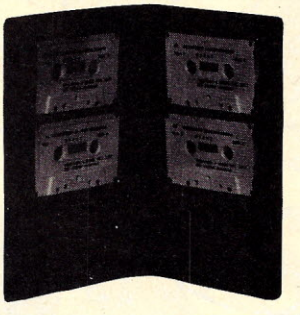
2161 SHATTUCK AVE.
Berkeley, CA 94704
TEL: (415) 848-8233

NEW SYBEX BOOKS

C201 - MICROPROCESSORS
420pp. 150 illustrations **$9.95**

C207 - MICROPROCESSOR INTERFACING TECHNIQUES
350pp, 320 illustrations **$9.95**

HOME-STUDY COURSES ON CASSETTES



Each course includes a special course book, plus audio cassettes, which can be played on a regular cassette player.

LEVEL 1: BASIC INTRODUCTION (no previous knowledge required).

S1 - INTRODUCTION TO MICROPROCESSORS: the hardware. What microprocessors/microcomputers are. The other components. Applications. (2½ hours) **$29.95**

S2 - PROGRAMMING MICROPROCESSORS: the software. How to program microprocessors, at the assembler level. (2½ hours) **$29.95**

LEVEL 2: INTRODUCTION (basic engineering knowledge required).

SA1 - MICROPROCESSORS: the components. How to interconnect a system. How to apply it. Basic programming. The most comprehensive overview (14 hours) **$59.95**

LEVEL 3: SPECIALIZED

SB7: MICROPROCESSOR INTERFACING TECHNIQUES: from keyboard to floppy disk, including CRT, cassette, TTY. The assembly of a complete system (7 hours) **$49.95**

SB5: BIT-SLICE: assembling an actual bit-slice processor, including the design of the control unit (7 hours) **$49.95**

SB3: MILITARY MICROPROCESSOR SYSTEMS: a comprehesive survey of the techniques and systems available (7 hours) **$49.95**

All "A" and "B" courses are also available on **VIDEO-CASSETTES** *(to be played on your TV).*

RECOMMENDED TEXTS:

C201 - MICROPROCESSORS (Rodnay Zaks), 420pp **$9.95**
C207 - MICROPROCESSOR INTERFACING TECHNIQUES (Lesea - Zaks), 350pp **$9.95**

DISTRIBUTION / TRANSLATION INQUIRIES INVITED

TO ORDER

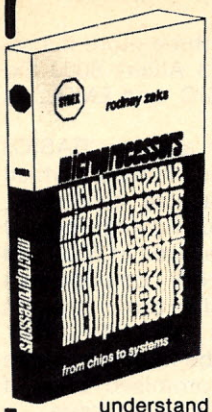
- **BY PHONE:** call (415) 848-8233
  BankAmericard/Mastercharge accepted
- **SHIPPING:** no charge when payment included.
  ADD: $1.00/book for fast shipping.
- **TAX:** in California, add sales tax.
- **OVERSEAS:**
  SYBEX-EUROPE, 313 rue Lecourbe
  75015 - PARIS, France Tel:(1)8282502

SYBEX

2161 Shattuck Ave.
Berkeley, Ca 94704
Tel: (415) 848-8233

NAME _____ POSITION _____
COMPANY _____
ADDRESS _____
CITY _____ STATE/ZIP _____
☐ Send me: ☐C201 ☐C207 ☐ Other: _____
☐ Payment enclosed ☐ Bill company ☐ Bill me
ADD $1.00 / BOOK FOR FAST SHIPPING
☐ charge my ☐ Visa ☐ Master charge
☐ Number _____ Exp date _____
Signature _____ ☐ Send catalog.

CC 78/3



MICROPROCESSORS
FROM CHIPS TO SYSTEMS
Rodnay Zaks
420pp, 150 illustr, ref C201 **$9.95**

Our Bestseller. This book is the result of the author's experience in teaching microprocessors to more than 2000 persons. It presents a comprehensive introduction to all the aspects of microprocessors, from the components to the assembly of a system. The difficulty of each chapter is graduated from the basic concepts to the actual technical details. It is read by students, technicians, managers, engineers, educators, doctors and by all those who wish to understand rapidly and efficiently all the important aspects of microprocessor use, selection, or application.

CONTENTS: *Fundamental Concepts... Internal Operation of a Microprocessor... System Components... Comparative Microprocessor Evaluation... System Interconnect... Microprocessor Applications... Interfacing Techniques... Microprocessor Programming... System Development... The Future...*

MICROPROCESSOR INTERFACING TECHNIQUES
Austin Lesea, Rodnay Zaks
350pp, 320 illustr, ref C207 **$9.95**

FROM KEYBOARD TO FLOPPY DISK: All the basic concepts and techniques for assembling a complete microcomputer system are presented in detail, with over 320 illustrations or diagrams, including teletype, keyboard, floppy disk, CRT display, analog / digital interfacing. One chapter is dedicated to the bus standards, including S100, and IEEE488, and one chapter to testing and debugging.

CONTENTS: *Introduction... Assembling the Central Processing Unit... Basic Input-Output... Interfacing the Peripherals... Analog Conversion... Bus Standards... Case-study: a 32-channel Multiplexer... Digital Trouble-Shooting... Conclusion-Evolution.*

DISTRIBUTION / TRANSLATION INQUIRIES INVITED

ALSO AVAILABLE: **MICROPROCESSOR ENCYCLOPEDIA**
- VOL I: 8-BITS, ref E8 **$15.00**
- VOL II: BIT-SLICE, ref E5 **$15.00**

TO ORDER
- **BY PHONE:** call (415) 848-8233
  BankAmericard/Mastercharge accepted
- **SHIPPING:** no charge when payment included.
  ADD: $1.00/book for fast shipping.
- **TAX:** in California, add sales tax.
- **OVERSEAS:** 66 French Francs. Write to:
  SYBEX-EUROPE, 313 rue Lecourbe
  75015 - PARIS, France Tel:(1)8282502

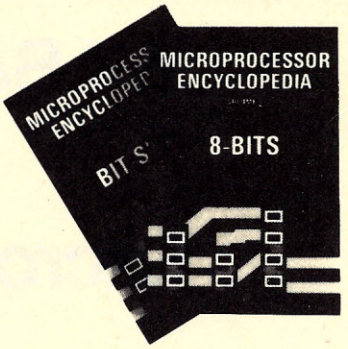
SYBEX

2161 Shattuck Ave.
Berkeley, Ca 94704
Tel: (415) 848-8233

NAME _____ POSITION _____
COMPANY _____
ADDRESS _____
CITY _____ STATE/ZIP _____
☐ Send me: ☐C201 ☐C207 ☐ Other: _____
☐ Payment enclosed ☐ Bill company ☐ Bill me
ADD $1.00 / BOOK FOR FAST SHIPPING
☐ charge my ☐ Visa ☐ Master charge
☐ Number _____ Exp date _____
Signature _____ ☐ Send catalog.

CC 78/3

MICROPROCESSOR ENCYCLOPEDIA



- **VOLUME I : 8 BITS** (ref E8) **$15.00**
- **VOLUME II : BIT-SLICE** (ref E5) **$15.00**

Each volume contains reprints of the complete data sheets for all the major microprocessors, as well as selected application notes. It allows easy comparison of the characteristics, at a glance. An efficient tool for evaluating/comparing/using microprocessors.

CONTENTS:
- **E8:** *AM9080, AMI 6800, EA 9002, Fairchild F8, GI LP8000, Intel 8080A, Intersil 6100, Mos Tech 650X, Mostek 3880, 3850, M6800, NS 8080A, SC/MP, RCA CDP 1802, PPS-8, Signetics 2650, TMS 8080, Western Digital MCP 1600, Zilog Z-80.*
- **E5:** *AMD 2901, Fairchild Macrologic, Intel 3000, MMI 5701, Motorola M2900, M10800, NS IMP, Signetics 8X02, 3000, TI SBP 0400.*

OTHER PUBLICATIONS:
- C201 - MICROPROCESSORS, by **Rodnay Zaks.** The basic text (420pp). Our bestseller **$9.95**
- C207 - MICROPROCESSOR INTERFACING TECHNIQUES, by **Austin Lesea** and **Rodnay Zaks,** 350pp **$9.95**

DISTRIBUTION / TRANSLATION INQUIRIES INVITED

S1 - Introduction to Microprocessors: book + cassettes..**$29.95**
S2 - Programming Microprocessors: book + 2 cassettes..**$29.95**

TO ORDER
- **BY PHONE:** call (415) 848-8233
  BankAmericard/Mastercharge accepted
- **SHIPPING:** no charge when payment included.
  ADD: $1.00/book for fast shipping.
- **TAX:** in California, add sales tax.
- **OVERSEAS:**
  SYBEX-EUROPE, 313 rue Lecourbe
  75015 - PARIS, France Tel:(1)8282502

SYBEX

2161 Shattuck Ave.
Berkeley, Ca 94704
Tel: (415) 848-8233

NAME _____ POSITION _____
COMPANY _____
ADDRESS _____
CITY _____ STATE/ZIP _____
☐ Send me: ☐C201 ☐C207 ☐ Other: _____
☐ Payment enclosed ☐ Bill company ☐ Bill me
ADD $1.00 / BOOK FOR FAST SHIPPING
☐ charge my ☐ Visa ☐ Master charge
☐ Number _____ Exp date _____
Signature _____ ☐ Send catalog.

cc 78/3

CIRCLE 128 ON READER SERVICE CARD

A Creative Computing Software Profile...

An Evaluation of an Extended BASIC, an 8K BASIC, and a Micro-APL

Steve North

In this review, we'll take a look at three more microcomputer languages: Microsoft (MITS Altair) 8080 Extended BASIC, SWTPC 6800 8K BASIC, and EMPL, an 8080 micro-APL.

The BASICs that have been known as Altair BASICs were written by a software house called Microsoft, which is now making these products available separately. Microsoft has a reputation for producing high-quality software, including the 8080 BASICs, as well as 6800 and 6502 BASICs, an 8080/Z-80 FORTRAN IV compiler, with more on the way. The Microsoft BASICs are in use in a wide number of machines, such as the MITS Altair, Commodore PET, and the soon-to-be-released Radio Shack Level II BASIC. Incidentally, the promised review of Microsoft FORTRAN IV will appear in a future issue.

SWTPC 6800 BASIC is also in use on a number of 6800-based systems, including the SWTPC 6800 computer, the MSI 6800 and Sphere computers (for those unfortunate enough to have bought one).

EMPL, a micro-APL, probably doesn't deserve to be graded on the same criterion as these other software products, since it is relatively low in cost and definitely an amateur effort, but we felt that it was interesting. Special thanks to Craig Finseth (author of "A Taste of APL" which appeared in the July-August 1977 issue of *Creative*) for helping out quite a bit with the EMPL evaluation.

In a future issue, we'll try to take a look at Microsoft FORTRAN IV, OPUS/ONE, and TDL Super BASIC Version 3.0. D. L. Dotson of Toledo, Ohio, sent an evaluation of TDL Super BASIC Version 2.1 which indicates that this is an excellent BASIC (the results of the SIDES3 benchmark program (Nov.-Dec. 1977 *Creative* were spectacular). However, a call to TDL revealed that Version 3.0 will include even more features and faster arithmetic functions, so we'll hold off on that. As of this writing, Version 3.0 is starting to be released.

There is a question as to whether we should review software products that are highly hardware dependent. For instance, we really don't want to review APPLE BASIC at this point, because the BASIC is in ROM and depends on certain hardware features of the computer. You can't take APPLE BASIC and just load it in any 6502 machine. In making a decision as to whether to purchase a Sol-20 or an APPLE II, there's a lot more to consider than whether one BASIC uses another two seconds to do a FOR loop. On the other hand, most hardware manufacturers realize that it's smart to tie their software into their hardware, to discourage software thievery. Additionally, manufacturers with a strong hardware background and with weak software programs have recently started purchasing systems software from a software house that may also have licensed the same software to a competitor. In a case like that, we *would* want to review the software package. (This has occurred with the Microsoft FORTRAN IV, in particular.) Some feedback from you readers on what kind of software *you'd* like to see reviewed would be appreciated.

Just for fun, I ran one of the benchmark programs, AHLDIG. F4 (Sep-Oct. 1977 *Creative*), in FORTRAN IV level G, on the New Jersey Educational Computing Network's IBM 370/168. This program normally takes an hour or more to run partway using a good microcomputer BASIC interpreter. Under FORTRAN IV level G, the compiled code took 2.2 seconds of CPU time to load and run to completion. (0.5 second of CPU time was used to compile the program, and 1 second of I/O time was used, but we'll ignore this since the benchmark is supposed to be purely computational). This of course proves beyond a doubt that the ideal personal computer is an IBM 370/168 running FORTRAN IV level G (Consult your IBM rep for details!)

MITS Extended BASIC 4.0

Author: Microsoft.
Size: 14.7k
Price and Availability: $150 for owners of an Altair 8800 with 16K of memory and an I/O board, $350 for others. MITS, Inc., 2540 Alamo S.E., Albuquerque, N.M. 87106.
Reliability: We found one bug in MITS Extended BASIC 4.0. When the automatic line-numbering feature is in use, and you type your own line number (perhaps by accident) and then certain statements, BASIC does strange things to the line, and goes away if you attempt to RUN this program. However, Microsoft has already released version 4.1 of this BASIC, and we suspect that they caught this minor problem.
Documentation: As we've mentioned before, MITS put all the documentation on 4K, 8K, and Extended BASIC into one manual, which confuses things for the user who wishes to find information on only one version of BASIC. The manual is complete and contains examples. However we don't quite understand why they chose to begin the manual with a discussion of RENUMber and AUTO. The source code is not available.
Speed: AHLDIG: 22 minutes to Step 33, 60 minutes to Step 55. With use of integer variables and division, the time to Step 33 was cut to 19 minutes. SIDES3: 48 seconds.
Features:

Commands: AUTO (automatic line numbering), CLEAR (with an expression, sets string space), CLOAD (loads arrays and loads and verifies programs from audio cassette), CONT (continue program execution), CSAVE (save array or program on cassette), DELETE (deletes portion of a program), EDIT (allows modification of a program line), LIST, LLIST (LIST on line printer), NEW, NULL (sets number of nulls printed after a CR/LF), RENUMBER, RUN, TRON and TROFF (control the program tracing feature).

Statements: CONSOLE (allows changing assignment of the console I/O device). DATA, DEF, DEFUSR (define the address of a machine-language subroutine), DIM, END, ERASE (eliminates an array—think of as the opposite of "DIM"), ERROR (forces an error with the code specified as the argument for user-defined errors), FOR, GOTO, GOSUB, IF...GOTO, IF...THEN, IF...THEN...ELSE, INPUT, LET, LINE INPUT, LPRINT (prints on line printer), LPRINT USING (PRINT USING on line printer), NEXT, ON ERROR GOTO (branches to specified line on error condition), ON...GOTO, ON...GOSUB, OUT (outputs to specified 8080 I/O port), POKE (stores a value in an absolute memory location), PRINT, PRINT USING, READ, REM, RESTORE, RESUME (resume program execution after an error—this is primarily for use in returning from an error-trapping routine), RETURN, STOP, SWAP (exchanges two variables), WAIT (waits for specified condition at an I/O port), WIDTH (sets terminal width).

Variables: Any set of up to eighteen characters is allowed as a variable name (provided there are no embedded BASIC keywords or special characters), but BASIC looks only at the first two characters in the variable name. MITS BASIC permits three types of variables: integer (0 to + 32767), single precision (+1.70141E38 to +2.9387E-38), and double precision (sixteen digits). Variable names that represent integers are postfixed with a % (as in D%), single-precision variables are postfixed with a ! (as in F!) and double-precision variables are postfixed with a # (as in R#). This can also be done with arrays (for instance, A%(2,3)). However, rather than throw all these special symbols around, it is much easier to declare variable types at the start of the program, using DEFINT, DEFSNG, and

DEFDBL. DEFINT A indicates that all variables beginning with A are integer variables. DEFDBL Q-Z indicates that all variables beginning with letters Q through Z are double precision. If you don't want to bother with all this single/double-precision stuff, BASIC will default to single precision, making the use of the ! symbol for single precision rather needless. The fly in the ointment here is that the transcendental functions do not have double-precision accuracy.

Actually, there is one instance where BASIC does not default to single precision. When you type in a variable with more than six digits, Extended BASIC says, "I'm sure that's a double-precision variable, so I'll put a # after it." If you entered 10 P=3.14159275, then BASIC will list this back as 10 P=3.14159275#. What does this mean? That Extended BASIC has made your program very un-transportable.

Octal and hexadecimal literals are also permitted. &(digits) or &0(digits) is used for octal literals. &H(digits) is used for hexadecimal literals.

Functions: +, -, *, /, ↑, \ (integer division), ASC (returns ASCII code of string), ATN, CINT, CNG, and CDBL (for converting variables from one type to another), CHR$ (converts ASCII code to character string), COS, ERL (line at which last error occurred), ERR (error code of last error), EXP, FIX (returns truncated integer part of argument), FRE (returns the number of free bytes of memory if argument is numeric, or number of free bytes of string space if argument is a string, HEX$ (converts argument to hexadecimal representation), INP (reads a byte from an 8080 I/O port), INSTR (searches for the occurrence of one character string within another beginning at a given character position), INT, LEFT$ (takes the left part of a character string), LEN, LOG, LPOS (position of the phantom line printer print head), MID$ (takes middle part of a character string), OCT$ (converts argument to octal digits), RND, POS (position of print head), RIGHT$ (takes right part of a character string), SGN, SIN, SPACE$ (creates a string of spaces), SPC (prints spaces on terminal), SQR, STR$ (converts argument to a string of digits), STRING$ (converts ASCII code to multiple-character string), TAB, TAN, USR, VAL (converts a string of digits to number), VARPTR (returns the absolute memory address of the variable).

User-defined Function: User-defined functions may have more than one argument and may be string functions as well. However, only a simple definition is permitted (not the multi-line DEC BASIC+ style definition). Arrays: Multi-dimensional arrays are permitted. Machine-language Subroutines: The DEFUSR statement is used to establish the addresses of up to ten user-defined functions (USR0 through USR9). For instance, DEFUSR1=20000 would indicate that the address of USR1 is 20000 decimal. To call this routine you would write variable=USR1 (argument). The machine-language subroutine must call subroutines within BASIC itself to pass parameters.

Character Strings: Extended BASIC has a rather complete set of character-string processing features, including string scalars and arrays. In addition to the functions listed above, it is possible to put MID$ on the lefthand side of a LET to insert characters in the middle of a character string. By using DEFSTR (similar to DEFINT etc.) you can force a numeric variable name to refer to a character-string variable, which could make program writing and debugging a bit tricky. The "+" symbol is used to concatenate strings.

Formatted Print: Extended BASIC has a PRINT USING statement. Specifications include:

- # for a digit
- ° for a decimal point
- + sign (at the beginning or end of a field)
- - sign (same as +, but suppresses positive sign)
- ** asterisk fill of leading spaces
- $$ dollar-sign fill of rightmost leading space
- **$ combination of above
- causes printing of comma in appropriate place
- exponential format

If the number to be printed will not fit inside of the PRINT USING field, then it is printed in normal BASIC format with a leading %. The number of digits may not exceed 24.

Editing Functions: The EDIT command may be used to modify an existing program line. Using a pointer, one may move forward in a line, insert or delete characters, search for a character in a destructive or non-destructive search, etc. The EDIT feature may also be used on the last line entered, by typing control-A. When a syntax error is found in the BASIC program, BASIC automatically enters the EDIT mode. This may be handy for the experienced, but it can only confuse beginners. Nevertheless, a very handy feature.

External Files: Only programs and arrays (no data files, per se).

Error Messages: 23 error messages with complete descriptions. For instance, RETURN WITHOUT GOSUB IN LINE 1270, rather than RG ERROR IN LINE 1270. Extra Stuff: Extended BASIC not only permits multiple statements per line, but multiple lines per statement! By typing a linefeed, you cause BASIC to begin a new physical line, but one logically attached to the preceding line. However this feature should be turned off during INPUT statements, which it isn't. For instance:

```
10 FOR X=1 TO 10 (linefeed)
PRINT X (linefeed)
NEXT X (return)
```

BASIC also permits use of ' in place of REM. Logical (boolean) functions include AND, OR, NOT, XOR, EQV, and IMP. BASIC has an initialization dialog that permits deletion of trig functions to free more memory. At initialization time, MITS Extended BASIC looks at the sense switches to determine what type of I/O interface to use (handy if you have a MITS interface, not handy if you don't). Control-O controls the printing of output. Control-S and Q are used to pause program execution and to continue. Control-I tabs over to the next logical stop. . denotes the current line number. Brackets may be interchanged with parentheses.

User Comment: Except for some superficial changes (bug fixes, multi-line user-defined functions, and perhaps a REPLACE command), Extended BASIC is probably as far as an interpretive microcomputer BASIC can be taken. If you have enough money to buy Extended BASIC and the memory to support it, then MITS Extended BASIC is what you want. There might be some debate as to whether Extended BASIC has in fact too many frills for the average user, or has more application in an educational environment than for the hobbyist.

We would nevertheless like to mention that we wish BASIC wouldn't play funny games with the source code by postfixing a # to indicate double precision. Of course, if you don't enter more than six significant digits, you're OK, but an interpreter should NEVER modify the user's program. After looking at Extended BASIC, you begin to wonder how much further microcomputer BASICs can be taken. The next logical step will be microcomputer PL/I, ALGOL, or some other structured language.

Southwest Technical Products 8K BASIC Version 2.0

Author: Robert Uiterwyk
Size: 7.9k. Requires MIKBUG ROM monitor.
Price and Availability: $9.95 for Kansas-City cassette from Southwest Technical Products Corp., 219 W. Rhapsody, San Antonio, TX 78216.
Documentation: The 27-page manual describes all statements, commands, and functions, with frequent examples. Although the source code is not available, at least a description of the memory map and "useful locations" are provided.
Reliability: In general, SWTPC 8K BASIC is very solid. However a friend recommended that we try

```
10 DEF FNA(X)=FNB(X)
20 DEF FNB(X)=FNA(X)
30 GOTO 10
```

which of course loops infinitely. However, if you break the program and type PRINT FNA(1), BASIC responds with an unmatched parenthesis in line 30 error message. No big deal.
Speed: AHLDIG: 85 minutes to Step 33, 205 minutes to Step 55. SIDES3: 417 seconds.
Features:

Commands: APPEND (appends to program buffer from cassette or paper tape), CONT (continue program execution), DIGITS (sets number of digits printed after decimal point), LINE (sets terminal width), LIST, LOAD (loads program from cassette or papertape), NEW PATCH (calls MIKBUG, the system monitor) PORT (used to direct I/O to a particular terminal), RUN, TRACE ON and TRACE OFF (control program tracing feature). Statements: DATA, DEF, DIM, END, FOR, GOSUB, GOTO, IF...THEN, INPUT, LET, NEXT, PRINT, POKE, READ, REM, RESTORE, RETURN, STOP.

Statements: DATA, DEF, DIM, END, FOR, GOSUB, GOTO, IF...THEN, INPUT, LET, NEXT, ON...GOSUB, ON...GOTO, PRINT, POKE, READ, REM, RESTORE, RETURN, STOP.

Variables: A-Z and A0-Z9. Precision is + 1.0E-99 to + 9.99999999E+99. Nine-digit accuracy is always carried internally.

Functions: +, -, *, /, ↑ ABS, ASC (returns ASCII value of character string), ATAN, CHR$ (converts ASCII number to character), COS, EXP, INT, LEFT$ (takes left part of character string), LEN, LOG, MID$ (takes middle part of a character string), PEEK, POS (position of print head), RIGHT$ (takes right part of character string), RND, SGN, SIN, SQR, STR$ (converts number to string of digits), TAB, TAN, USER, VAL (converts string of digits to a number).

User-defined Functions: Yes, with one argument permitted.

Arrays: One or two-dimensional arrays are permitted. However subscripts may not exceed 255 (most other microcomputer BASICs permit something reasonable, over 1024. Also, arrays start at 1, not 0 (as in X(1,1)—thus X(0,0) is illegal). Array names are A-Z.

Machine-Language Subroutine Interfacing: The USER function is used to call machine-language subroutines. The location of the subroutine must be stored at locations 67 and 68 in BASIC. The argument of the USER function is stored in the seven bytes beginning at the location pointed to by locations 5D and 5E.

Character Strings: Strings may contain up to 32 characters (you may sometimes wish it was more, but this is a great improvement over Version 1.0, which only allowed 18 characters in a string). The string-

handling functions should be adequate for most applications. + is used for string concatenation.

Formatted Print: None.

Editing Functions: None.

External Files: SWTPC 8K BASIC doesn't have any formal file-handling features. However, since you can INPUT and PRINT to any port by placing a #(port number) after the keyword (as in INPUT #5,X) it is relatively easy to "PRINT" and "INPUT" to and from the cassette unit.

Error Messages: 28, specified by two-digit error codes.

Extra Stuff: Because SWTPC 8K BASIC does permit PRINT # and READ # to direct I/O to or from a specific port, it is especially suited for multi-terminal games.

User Comment: SWTPC 8K BASIC is probably the only widely-used 6800 8K BASIC. It has no frills and is rather slow, but this is more than offset by its extremely low price and high reliability. We would like to commend SWTPC for providing its customers with inexpensive, useful software. Incidentally, if you're still using SWTPC 8K BASIC Version 1.0 or 1.01, you may wish to upgrade to Version 2.0, since it has a number of small bug fixes, the ATAN function, faster functions and line searching, and some other finishing touches.

EMPL

Author: Erik Mueller.

Price and Availability: $10 for a Tarbell cassette, from Erik Mueller, Britton House, Roosevelt, NJ 08555.

Size: 5.4K

Reliability: Craig Finseth suggested a bug, which was in APL*CYBER (Control Data's old APL). He says, "You basically build up something in the stack and bomb out of a function. You then delete one of the functions in the call chain, and continue. Interesting things tend to happen. Most APLs (including CDC's new one, APLUM) catch this and issue a SI DAMAGE error." However, EMPL did not. Instead it sometimes thought the function was still there. Although it appeared that the function could be edited correctly, there were phantom lines from the old (deleted) function.

The other problem is that EMPL never checks for numeric overflow. A number that overflows suddenly becomes negative (since numbers are 15 bits and a sign bit, the carry-out of the next to high-order bit is placed into the sign bit).

Documentation: A 23-page manual is included. The manual contains customization details, a description of the language, and some examples. Unfortunately, this manual will not teach you to use EMPL if you don't know APL. It is also probably very difficult to learn EMPL using an APL manual. This suggests that it is extremely helpful to know APL before using EMPL. The source code for EMPL is not available. Speed: Because of the intrinsic differences in BASIC and APL language structures, the timing comparisons were not run. One can't simply take a BASIC program and convert it line-by-line to APL, as you would in converting BASIC to FORTRAN. However, EMPL seems to be reasonably fast; that is, it compares with other tiny languages using integer math.

Features:

Commands: (All commands are preceded by the character ')'.) CLEAR (clear workspace), FNS (display function names), VARS (display variables), PUR (clear state indicator), ER obj (erase object), SCN (clear CRT screen), SI (display state indicator), STOP (returns to execution mode), QUIT (returns to user's monitor program).

Variables: All values are 16 bits (range 0 to + 32367). Only scalars and vectors are available.

Operators:

Monadic Scalar Operators: - (negate), ! (absolute value), & (not) ? (random).

Monadic Mixed Operators: \ (vector of consecutive integers), ↑ (length of vector).

Dyadic Scalar Operators: +, -, *, /, ' (minimum), " (maximum), . (remainder of division), < , >, <=, >=,=, # (not equal).

Dyadic Mixed Operators:, (concatenate), [(elements of vector).

Composite Mixed Operators: % (operation performed left to right on all elements of variable).

Special Operators and Characters: := (assignment), $ (print string), "TEXT" (print literal text), %:(executes a string), =: (branch to), Y(X) (indexed variable assignment), @ (numeric I/O), $ (string I/O), 'TEXT' (string vector), (X) (parenthesized expression), -C (constant negation).

Definition Mode Commands: @ (list), $ (renumber), & (return to execution mode).

Function Definitions: &F (niladic), &R: = F B (monadic), &R:= A F B (dyadic).

Extra Stuff: In order to implement characters in the APL character set using ASCII, EMPL requires that control characters be entered for certain operators. The control characters are echoed as two printing characters. For example, to enter := (assignment) you type control-I. Typing := separately won't be recognized by EMPL.

User Comment: Evaluating a language like EMPL is more difficult than evaluating a BASIC interpreter. EMPL is a somewhat warped subset of APL but it is the only 8080 APL available today. Craig Finseth had some interesting comments on EMPL. He said, "I feel that EMPL's use of ASCII is, in general, a good idea. While his selection of characters is far from ideal, it is a lot better than saying "too bad" when someone doesn't have an APL terminal."

However, Craig goes on to note, "Finally—and this is a personal opinion—Mr. Mueller has missed the point of APL. APL is not a syntax or order of evaluation, unusual as they may be. Neither is it a large collection of number-crunching primitives, ranging from identity to matrix inversion. Rather, it is built around a concept, that of the n-dimensional array. If you have APL without arrays, or (as with EMPL) one with only vectors, there is very little reason to use APL at all. Where APL's power comes in is with the array-handling and processing: grade up and grade down (sort), rotate, transpose, selection and expansion, reduction and scan, inner and—most importantly—outer product. These functions and operators, together with the n-dimensional array, allow APL to be a language that expresses the algorithm more as logical chunks than physical ones."

Craig also found it impossible to convert his prime-number program from APL to EMPL. In other words:

- If you don't know APL, then EMPL will be very difficult to learn;
- If you do know APL, then EMPL will make you wish you had the real thing.

However, EMPL is an interesting experiment. It won't be long before full APLs are available for microcomputers. A group at Texas A&M is doing promising work in this area. In their version, a standard ASCII keyboard (with control characters) is used for input, and a Processor Tech VDM-1 (with displayable control characters) for output. Also, in the small print at the bottom of one of its ads, Microsoft announced its intentions to release an 8080 APL this year, which will no doubt be a high-quality product. The problem of mashing the APL character set to fit into ASCII will eventually be of little importance, since video displays with user-programmable character ROMs or software-controlled character sets are beginning to appear on the market. ■

The ABCs of Microcomputers

Steve North

What the Beginning Computer Hobbyist Needs to Know

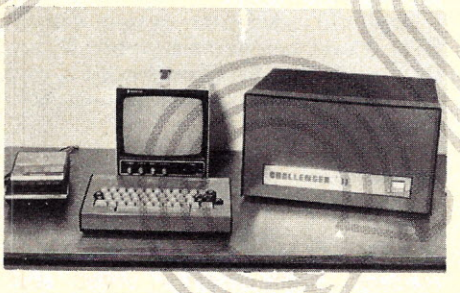
Perhaps you're a novice to the personal computing field and aren't quite clear about what a complete personal computer system really is. Or perhaps you're a teacher who is considering the use of microcomputers for educational applications. Which microcomputer is best for you? Upon what features should you base your decision? How can you act smart and ask the right questions when you walk into your local computer store? In this article we'll try to cover the important features of a good general-purpose microcomputer. Remember though that if you have a very specific or unusual application, you're not going to be looking for the same things as someone who just wants to play Star Trek. The areas we'll look at are:

Central Processing Unit
Front Panels
Bus Structure
Memory (RAM and ROM)
Input/Output Interfacing and Peripherals
Software

Central Processing Unit

The CPUs in microcomputers are based on microprocessors, which fit many of the components of the CPU on a single integrated-circuit chip. In comparison with large computers, microprocessors are slow. They do not have any built-in floating-point math capabilities. Most microprocessors used in personal computers are 8-bit processors, although some 16-bit operations are also provided. A few 16-bit processors are finally making their way into the hobby-computer market, but they are still relatively high in price compared with 8-bit processors. We're not trying to say that microprocessor-based systems aren't computers, but merely that they don't perform like IBM 370s. They are still extremely powerful tools.

While there are scads of different microprocessors, only a few are commonly used in personal computing systems. It is best to stick to one of the common ones, since this will make it possible to use software other people have written, rather than have to do it all yourself. The most common microprocessors are the 8080, 6800, Z-80, and the 6502. Each of these microprocessors speaks a different machine language. You can't take a 6800 machine-code program and run it on an 8080. (We should point out that the Z-80 instruction set is a superset of the 8080 instruction set, so all 8080 programs will run on the Z-80 but not all Z-80 programs run on the 8080. Also, the 6800 and 6502 instruction sets are very similar). If you plan to do much assembly-language programming, where it is necessary to understand the machine language of the processor, then you should first compare the instruction sets of the various microprocessors to see which you prefer. Some people claim that certain microprocessors resemble certain large-scale computer systems; for instance, the 6800 is similar to a PDP-11, and the Z-80 has some 360-ish instructions. While these similarities may be slightly contrived, you might also take this factor into consideration if you are familiar with a large-scale computer.

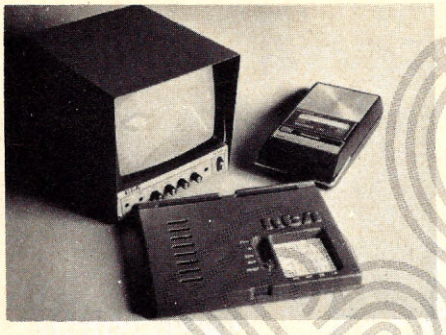


Ohio Scientific's Challenger II computer has a 4K RAM memory, and optional 8K BASIC in ROM.

A CPU contains a system clock that controls the speed of operations within a computer system. In general, most 8080 and Z-80 systems run at 2 MHz, while 6800 systems run at 1 MHz. However the 6800 is just as fast as the 8080 because it uses fewer clock cycles to do the same work. Some Z-80 systems now run at 4 MHz, which means that they can compute twice as fast as a 2-MHz system. However a clock that fast can also be too fast for other modules in the system, so take care if you choose a 4 MHz clock computer. Actually, most people don't run many computationally-bound programs on their personal computers, so clock speed is generally a minor factor.

Front Panels

A front panel is a console with switches and lights used to observe and control the internal activity of a computer. For instance, the front panel can be used to read or write memory data, start or stop the processor, etc. Programming with a front panel tends to be very slow and error-prone as well as boring. Once you have your computer talking to you with a terminal, you



RCA's low-cost COSMAC VIP computer-on-a-card, with a 2K RAM memory and a hex keyboard.

won't use the front panel much. So, for actual operation of a computer the front panel is not necessary and usually superfluous. On the other hand, front panels are useful for hardware debugging (if you plan to do much of that), and look impressive.

Bus Structure

The "bus" in a microcomputer serves as the electrical interconnection between modules (circuit boards) in the system. Some computers use a standard bus. That means that you can take a memory board, for example, from computer X and plug it into computer Y. There is nothing particularly special about a standard bus, except that several manufacturers have agreed to make their boards plug-compatible with it. (Actually, they didn't actually agree — they just decided that there was more money in making products for which there are more potential customers). This encourages lower prices and diversity because of competition. If your computer doesn't have a standard bus, then you'll have to depend on your particular computer manufacturer for new options when he's ready to release them, and at his prices.

The two standard busses are the "S-100" (Altair/IMSAI/Processor Tech/Poly 88/etc.) bus and the "SS-50" (Southwest Technical Products/Midwest Scientific Instruments/Gimix) bus. The S-100 bus is used in most 8080 and Z-80 systems. The SS-50 bus is used in 6800-based systems. There are many more boards for S-100 bus systems than there are for SS-50 bus systems, but the SS-50 bus appears to be increasing in popularity (a bus race).

If you're looking for a special-purpose computer system, or don't plan on much expansion, then bus structure won't make that much difference. For example, the Compucolor 8001 is an 8080-based color-graphics computer but not S-100. However if you're very interested in color graphics, that certainly shouldn't stop you from buying one!

Memory

Memory is measured in bytes (8 bits) and Kbytes (1024 bytes). Two types of memory are used in personal computer systems. One is called RAM, which stands for random-access memory. RAM can be written into, and read from, by the computer. This is the kind of memory used for most systems software (a BASIC interpreter, for instance) and program workspace (your Star Trek game written in BASIC). So the more RAM you have, the better, as long as you have a program big enough to use it. 8K may be sufficient for a start; 16K is more than enough for most applications. But there always seems to be at least one program you could try out, "If I only had another 4K of memory." Memory boards usually hold 4K, 8K or 16K of RAM. Remember that the memory requirements mentioned above refer to a general-purpose system running BASIC, and not a small microprocessor-based controller running a small machine-language program.

The other type of memory is ROM, which stand for Read-Only Memory. The computer can read information from this memory, but it can't store new information. So ROMs must be preprogrammed with a set of instructions. The program is often a simple monitor routine used to control basic operations of the computer. For instance, it might permit you to enter and dump memory locations, load a program, execute a program, etc. A program stored in ROM is always present, even when the power is turned off. A few computers come with BASIC programmed in ROM, so that as soon as you turn on the system, it is speaking in BASIC. In a system with no front panel, you must have a program in ROM that will permit you to control the computer. Even if you have a front panel, ROM eliminates the need for toggling in a short program every time the system is turned on. Typical monitors occupy 512 to 2048 bytes of ROM, while BASICs generally require 8K. Erasable, user-programmable ROMs are also available, so you can write your own programs and put them in ROM. If you decide to go this route, you will also need some means to program the ROMs.

Now our computer system has a CPU, a bus, and some memory. It can now run programs. But we need to get information in and out of the computer. This brings up:

Input/Output Interfacing

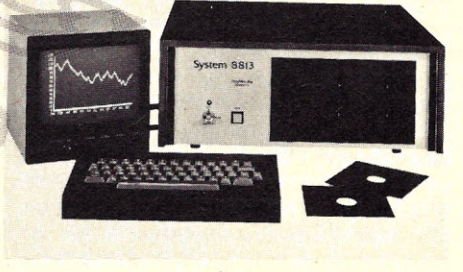
Special modules are usually needed so that your computer can send and receive data from terminals, papertape readers, audio-cassette units, and other devices. There are two types of general-purpose interfaces.

Serial I/0 is input and output performed with data transmitted as a stream of ones and zeros along a single path (wire). Serial I/O systems is in personal computers only require send, receive, and ground signals, though commercial serial data protocols are much more complicated. Teletypes, CRTs, and other terminals generally speak to serial interfaces.

There are also two types of serial interfaces: EIA RS-232 and current loop. These are merely different methods of sending the serial signal, so most serial interfaces permit you to use either one. Serial I/0 can also be done at different speeds, called "baud rates." This is also adjustable on the interface board to match the speed of the terminal.

Parallel I/0 is the other type of general-purpose interface. With this method, all the data is transmitted at once, along separate pathways. To send 8 bits of data, we use 8 wires (plus a ground wire for reference, and a strobe to signal that data is available).Parallel devices are usually keyboards, optical papertape readers, and other relatively simple devices.

But how do you save and load programs with your computer? And how do you talk to your computer if you don't want to buy an expensive terminal? There are special-purpose interface boards to handle these needs. For saving and loading programs, there are cassette interfaces. These boards permit data to be saved and loaded with an inexpensive audio-cassette recorder. Unfortunately, different audio-cassette interfaces are rarely compatible, and at last count there were at least a dozen different types. Audio-cassette interfaces range in speed (measured in bytes per second) and reliability. It is best again to stick with one of the most common types, so you can trade programs with other people and buy software sold commercially. The most common cassette standards are Tarbell (187 bytes/second), Kansas City (30 bytes/second), CUTS (120 bytes/second) MITS (30



PolyMorphic Systems' 8813 is at the high end of the hobby market, with a triple disk drive, BASIC interpreter, text editor, and assembler.

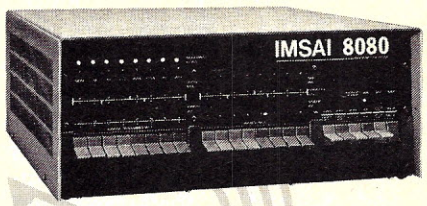
bytes/second) and TDL (120 bytes/second). Some of these standards are used only by owners of the manufacturer's system: The MITS cassette interface is used almost only by MITS Altair owners; the Kansas City Interface is used by SWTPC 6800 owners, and the TDL cassette interface is available only on the TDL System Monitor Board.

Special interface boards are also available to permit you to connect your computer to an ordinary television set and use that for output. These boards display 64 columns and 16 rows of characters in black and white, and have some other features. However they require more complicated software to use than a terminal, so if possible, try to get a system with the software to control this board in ROM.

For most people, audio cassettes serve as an entirely adequate means for saving and loading programs. However, audio cassettes are somewhat slow (you may have to wait a minute or two while loading a program) and they don't permit you to randomly access data. You can't tell your audio-cassette interface hooked up to a $39 cassette recorder, "Go find the 37th record on the file and when you get it, rewrite it with this information." It's just not possible. If you need to do high-speed, random-access data manipulation, then you may need to get a floppy-disk unit. A floppy disk itself is a circular piece of magnetic tape, several inches in diameter, which is inserted in a floppy-disk drive. The drive permits your computer to locate any data on any portion of the diskette. Floppy disks come in two sizes. The small size can hold about 90 Kbytes of data per diskette, the large size holds over 300 Kbytes of data. But be warned! While the diskettes are relatively cheap, the drives and interface in the computer can be rather expensive: $700 and up (kit) for the small drives, and $2000 and up for the larger drives. It is sometimes useful to have two drives, especially when copying a diskette, or updating a file. The difference between one drive and two is usually only a few hundred dollars. A floppy disk is definitely not a necessity and in fact most hobbyists do not have one. But they do increase the versatility and power of your computer system.

Software

This feature can not be underemphasized. To have a useful computer system, you must get software with the system. Look for: (1) a general-purpose monitor program,

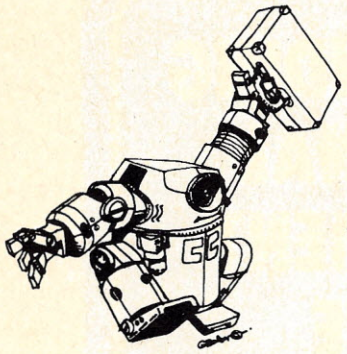


The IMSAI 8080 has the full complement of front-panel switches and lights, uses the S-100 bus, and has BASIC interpreters in 4K, 8K and 12K sizes.

(2) an assembler, and (3) a BASIC interpreter program. Other packages, such as text editors, are also very useful. Remember too that the software must be compatible with your hardware. In other words, BASIC should be able to load and save programs with your audio-cassette interface.

There is much more we could say about the components of a general-purpose personal computer system; it's difficult to cover all the important points in a book, let alone an article. We suggest then that before you make any snap decisions about which personal computer to buy, you read as much as you can, carefully consider manufacturer's literature, and visit your computer store (as well as any friends who may own personal computers). ■

Profile of a Smart Electronic Game...



Created to put fun into learning math, this teaching calculator and game machine is profiled by its designers.

APF Mathemagician

Joseph W. Willhide
Henry L. Viarengo

The regimen of drill and practice is very much a part of teaching mathematics at the primary-school level. Educators have long endeavored to make this a more palatable experience—first through the use of creative workbooks and more recently through use of CAI systems. As one might expect, each of these approaches has its own unique set of strengths and weaknesses. The workbook, besides being economical, provides what the authors would like to term "personal" learning—it belongs to the child, can be taken almost anywhere and can be worked with almost any time. It suffers, however, from being rather static and familiar. It may also suffer from inadequate feedback and rigidity of structure. CAI systems, on the other hand, tend to be very dynamic, engaging, and offer the ultimate in the implementation of the latest educational approaches to learning. They are, unfortunately, usually implemented on systems costing many tens of thousands of dollars. Hence such systems are owned and operated by institutions that seek to achieve a low cost per user by amortizing the cost of such a system over a large number of users. This leads to learning which the authors would like to term "institutional"—it tends to come in a fixed block of time on a fixed schedule and at a fixed location.

Technological advances in microcomputers coupled with mass-market economics of scale have led the authors to design a new consumer-electronics product that positions itself somewhere between the above two approaches. This product, the Mathemagician, is a portable, battery-powered, multi-functional teaching calculator and electronic game machine designed to introduce fun and excitement into the learning of mathematics for five- to twelve-year-olds. It is being manufactured and marketed by APF Electronics and appeared in major retail catalogs and stores last year. With its availability in the consumer marketplace, the Mathemagician qualifies as a "personal" learning aid. It has, however, many of the desirable attributes of the "institutional" CAI systems.

The Mathemagician is about the size of a portable magnetic-tape cassette recorder (5¼" x 8½"). It features a very large (0.3") numeric display which permits more than one viewer and a quality keyboard of the desktop-calculator type. The case comes with a carrying handle that can also be folded back underneath when in use, to give a desirable slope to the unit. A storage compartment is provided in the bottom of the case for the plastic overlays used for the games. Powered by easily-obtainable C-size batteries, it can be used anywhere and at any time. An optional AC adaptor is available.

The Mathemagician as a Teaching Calculator

As a teaching calculator, the Mathemagician has three basic modes—calculator, number-sentence checking and problem presentation. Calculations are performed in the calculator mode in a manner that is natural to a young child. That is, not only does the answer appear in the display, but also both numbers entered along with the mathematical operation symbol and equal sign appear, giving a complete number-sentence format. For example, if one wanted to subtract five from four, the keystrokes would be

[4] [-] [5] [=] [?]

and the display would read

4 - 5 = -1

Because of its emphasis on the earlier years of mathematical development, the Mathemagician uses remainders in division. Hence, in dividing thirty-five by two, the display would read

35 ÷ 2 = 17r1

The unit's display can accommodate two-digit operands and up to four-digit answers.

In the number-sentence checking mode, a complete number sentence is entered, followed by touching the question-mark key. If the number sentence being displayed is correct, then an indicator labeled "That's Right" lights; otherwise, the number sentence is corrected and an indicator labeled "My Answer" lights. Thus, for the entry

[1] [5] [X] [3] [=] [3] [5] [?]

the number sentence would be corrected and the display would read

15 X 3 = 45

with ■ MY ANSWER illuminated.

The problem presentation mode is the most powerful and flexible of all the teaching-calculator modes. It casts the learning and practice of the basic arithmetic operations into what might be called a series of games. These games can be played either by a single player or in a competitive two-player mode. They can be played in a play-against-the-clock mode in which responses must be entered within a specified response time, or played in a non-timed mode. Each game consists of ten rounds. A score is automatically

Joseph W. Willhide, Boston University; Henry L. Viarengo, Entrex; paper delivered at NAUCAL '77, Dearborn, MI, Nov. 3-5, 1977.

kept for the players(s) and displayed at the end of the game. In the case of two players, both scores are also displayed at the end of each round to foster a competitive spirit.

During each round, a player is given a problem of the type that was set up at the beginning of the game. For each problem a flashing question mark prompts the player to supply the missing part of the number sentence displayed. If the answer is correct, the "That's Right" indicator lights; otherwise, the "My Answer" indicator lights and the corrected answer appears in the display. Interest and excitement are heightened by the way each problem is dynamically built, one digit at a time, by moving the digits across the display from the right-hand side.

The level of problem difficulty can be adjusted over a wide range, allowing the unit to continually challenge, but not frustrate, the player. This is done by "programming in" the type of problem and difficulty-level during a setup phase prior to game playing. Some game setups are illustrated below.

[SEL] [7] [X] [___] [=] [NXT]
This sequence of keystrokes sets up a game to practice multiplying by seven. During the ten rounds of the game, all possible combinations of seven times a single digit would occur, in random order.

[SEL] [___] [___] [-] [___] [=] [NXT]
This setup generates a game whose rounds are all exercises in subtracting a single-digit number from a double-digit number

[SEL] [___] [+] [?] [=] [NXT]
This sets up a game in which the player(s) must complete a number sentence by entering a missing factor on the left-hand side of the equal sign.

Game setups, such as above, are explicitly given in the instruction book. Because of the unit's versatility, however, not all of the possible "games" can be covered in the instruction book. Many users will be able to see "the order of it all" and set up games not in the instruction book. In some cases these users will be teachers or parents working with the young players. In other cases, it will be the young players themselves, as part of the attractiveness of the Mathemagician is that it can serve as a vehicle for experimentation and self-discovery.

Mathemagician as an Electronic-Game Machine

The Mathemagician comes with six built-in, preprogrammed games that provide entertaining play while encouraging kids to learn about numbers. Each game has its own colorful plastic overlay that customizes the unit to that game.

Some of the games have been designed for the very youngest member of the family, while others will engage older children and adults. The games can be played by one or two players and have been designed to encourage the development of logical thinking as well as provide fun. The games, in ascending order of age appeal, are: NUMBER MACHINE, COUNTIN' ON, WALK THE PLANK, GOOEY GUMDROP, FOOTBALL and LUNAR LANDER. Each game has a unique number associated with it and is activated by the keystroke sequence [SEL] [Game#] [NXT]. ■

APF Mathemagician Games

David H. Ahl

In addition to being a highly versatile teaching calculator the APF Mathemagician has six games built in. Each uses a plastic overlay and, with the exception of Lunar Lander, can be played by one or two players.

Number Machine

This game is designed for very young children to help them develop number recognition skills. One of the digits 0 to 9 is displayed in a window for one second; the player must then key in the same number. As with teaching calculator functions, the score is given at the end of ten rounds. This game, like the others, may be played by one or two players. Two players alternate turns and their scores are displayed after each round at the extreme right and left side of the display.

Countin' On

In this game the Mathemagician will count by 1's or skip-count by 2's, 3's, 4's, etc. Each digit is displayed for one second. The player counts along, and keys in the number where the counting stopped. For example, after seeing 5-5-5-5-5, the player would key in 25. The calculator indicates whether or not the answer is correct and, if it is not, the correct answer is displayed. My 6- and 7-year olds liked this game when set to count by 5's. My 9-year old liked 2's and 5's but then soon realized that getting the correct answer with other numbers was simply a problem of multiplication.

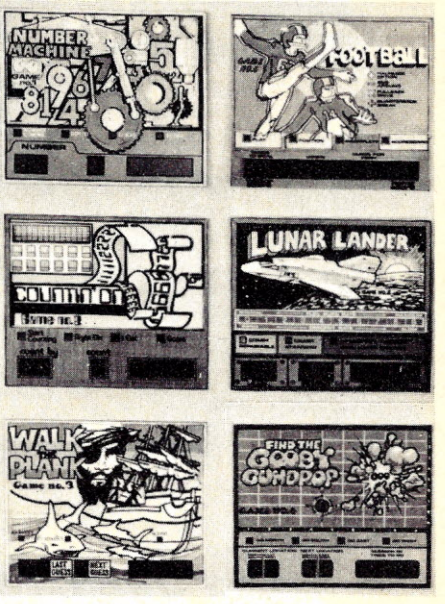
Walk the Plank

The Mathemagician picks a secret number from 1 to 9 and the player gets three tries to guess what it is. After each try, the player is given clues whether his guess is too high or too low. If the guess is right the word **Yes** will show in the display. If the player doesn't guess the number by the third try, **byby** will show as you go into "the drink."

My kids liked this game and soon realized that 1 and 9 were not good numbers with which to begin guessing. They haven't rigorously stumbled onto binary search and I'm not planning to tell them about it, but they're coming close. Of course, with only three guesses, it is impossible to always get the secret number even using binary search. Let's say the secret number is 1. Your first guess is half of the total interval or 5. Clue: smaller. Next guess is half the lower interval or 3. Clue: smaller. Third guess is either 1 or 2. If 2, you go byby.

Gooey Gumdrop

In this game the player must find a Gumdrop before it blows up. The Gumdrop is hidden in a nine-block by nine-block area of Gotham city. It takes two numbers to guess a location; the



Overlays for the six games that can be played with APF's Mathemagician.

first is North-South and the second East-West. After each guess you are given clues whether to go North, South, East or West. If only North lit up, it would indicate that you had found it in the East-West direction. You always start at 1,1. At the beginning of each round, you key in how many guesses you want. If you don't find the Gumdrop, the number of total guesses is added to your score; if you find it, only the number of unused guesses is added. A low score is best in this game. My kids quickly discovered that they could nearly always find the Gumdrop in four tries (if you remember the clues, you *always* can in four). The only objection I have to this game is that the North-South location is given first, unlike standard algebraic matrix notation in which the X coordinate is first followed by Y.

Football

This is a game of math practice. On each play you determine whether you want to do a +,-,x, or ÷ problem. All problems are formed with one-digit numbers. Answer correctly, and the ball moves toward the goal line the number of yards in the first digit of your answer. If the answer is wrong, there is no gain. The rules are like regular football; you must make 10 yards (a first down) in four plays or you lose the ball.

My kids occasionally got angry with this game after moving the ball from, say the 95-yard line to the 10-yard line and then getting four problems with answers in the tens or twenties which cause a loss of the ball. On the other hand they liked this game because it allowed plays at different math levels to play together. For example, my 6-year old used only addition problems, my 7-year old, addition and subtraction, and my 9-year old, all problem types.

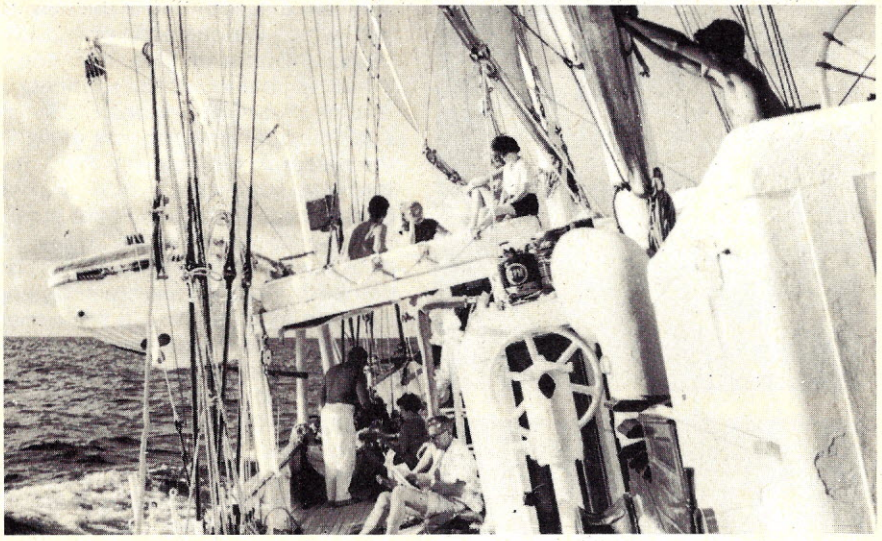
Lunar Lander

In this game you attempt to soft-land a LEM on the moon from 300 feet up. You have 99 units of fuel and start with a speed of 20 feet per second. You key in an amount of fuel, the rockets "burn" this and then tell you where you are. Come in too fast and you crash. Slow down too soon and you run out of fuel and crash.

A challenge to adults and a favorite computer game, this was a bit beyond my kids although they liked to fool around with it. To really achieve a good landing, you'll want to keep a pencil-and-paper record of previous trials.

Altogether I've found Mathemagician an excellent teaching calculator and game player that has provided many hours of educational fun for my kids. The APF Mathemagician is available from Sears and other department and specialty stores for $35 to $45.

Take off your shoes.



Hit the deck in shorts and a tee shirt. Or your bikini if you want.

You're on a leisurely cruise to remote islands. With names like Martinique, Grenada, Guadeloupe. Those are the ones you've heard of.

A big, beautiful sailing vessel glides from one breathtaking Caribbean jewel to another. And you're aboard, having the time of your life with an intimate group of lively, fun-loving people. Singles and couples, too. There's good food, "grog," and a few pleasant comforts...but there's little resemblance to a stay at a fancy hotel, and you'll be happy about that.

Spend six days exploring paradise and getting to know congenial people. There's no other vacation like it.

Your share from $265. A new cruise is forming now. Write Cap'n Mike for your free adventure booklet in full color.

Windjammer Cruises.

Name _____

Address _____

City _____ State _____ Zip _____

Phone _____

P.O. Box 120, Dept. 581 Miami Beach, Florida 33139

CIRCLE 125 ON READER SERVICE CARD

An exclusive interview with Jurg Nievergelt conducted by Karl L. Zinn at the NAUCAL 77 Conference, Dearborn, Michigan, Nov 77

"Programming Is Learned By Practice, Not By Listening"

Karl Zinn: Your presentation at NAUCAL 77 was received well by the audience and I am pleased you will share some of it with *Creative Computing* readers, and answer additional questions. Your automated course for teaching computer science on the Plato CAI system impressed me as highly significant for two reasons: a demonstration of the ways in which an interactive system such as Plato can be used, as well as its contribution to the teaching of computer programming. What has come of your work funded by NSF at the University of Illinois?

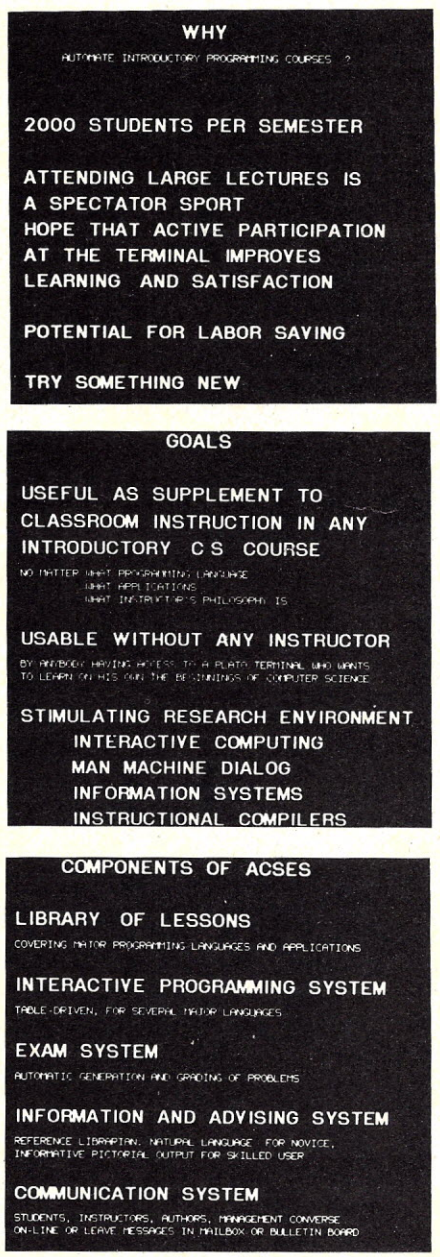
Jurg Nievergelt: We developed a system called ACSES, the Automated Computer Science Education System, from 1972 to 76. It was by far the biggest software project I ever was involved in - we generated about a million lines of code with an effort in excess of 20 man years. I set the goals high - I wanted to have the best automated course anywhere. And I figured that if we missed our goal by a factor of 2, we would still end up with a very respectable product. That's how things turned out, in my opinion.

KZ: What is the current status of ACSES?

JN: ACSES is now in routine use, carrying about 50% of the instructional load in the first CS courses at the U of I. 1500 students every semester spend 1 to 2 hours a week on Plato studying lessons on Fortran, PL/1, computer applications, and doing exercises on-line. From this point of view, ACSES has been a great success - my greatest worry during the 4-year period of hard development work was that this instructional system might not be used.

KZ: Developing a multifaceted curriculum of that size for an introductory computer science course is a very ambitious project. Are there any parts of it that did not work out?

Karl L. Zinn, Center for Research on Learning and Teaching, University of Michigan, Ann Arbor, MI.
Jurg Nievergelt, Institute for Informatics, ETH Swiss Federal Institute of Technology, CH-8092 ZURICH and Department of Computer Science, University of Illinois, Urbana, IL.



The instructional system ACSES — why and how.

All of the illustrations in this article were photographed directly from a Plato terminal. Unfortunately much of the outstanding resolution was lost in the many photographic steps between the original display and printing here in the magazine.

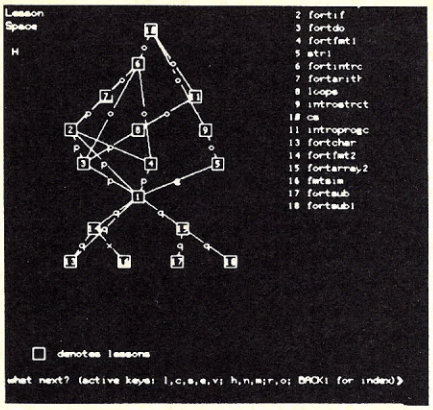
JN: I am disappointed at the slow pace with which various parts of ACSES are being introduced into our courses. What is being used routinely are routine tutorial lessons. The most sophisticated components of ACSES, the research projects and Ph.D. thesis that were supported by NSF, are hardly being used. The reasons are many.

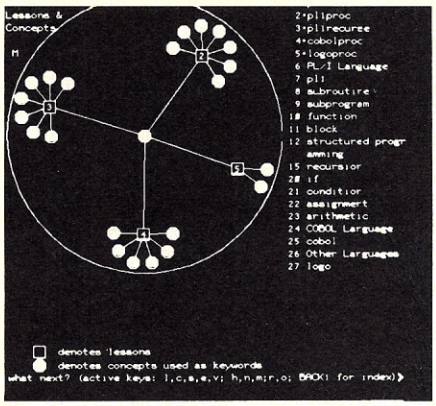
For example, the single largest piece of coherent code in all of ACSES is the compiler/interpreter system developed by Tom Wilcox. It is a monumental piece of work. It runs small programs in such different kinds of languages as Fortran, Cobol, PL/1, Pascal, Lisp, Snobol; with excellent facilities for diagnostics and error analysis at program entry- and run-time. For example, when an execution error has occurred, say overflow, the system engages the student in a dialog during which the program is executed backwards, tracing the values of the variables that caused the overflow. With the current load on Plato (often 500 simultaneous users), there is not enough memory and CPU power to run this interactive programming system, and so it is not being used. That's a pity, because it is superbly suited for programming instruction.

Then there is an information and advising system, csguide, that knows what instructional material is contained in ACSES, knows what a student has done so far on Plato, what he is supposed to do by what date, and on this basis advises him what he should be doing whenever he signs on. The guide talks to the user in terms of pictures that display the relationship among lessons in the library and computer concepts. We often assimilate graphic information much faster than text.

In the courses at Illinois, the instructor performs these functions, so there is little need for csguide. If ACSES should ever get used extensively at other schools, or for self-study, then csguide will become important.

We also have a fancy exam system, based on a library of "problem generators and graders". An instructor





The information system CSGUIDE responds to a user's query by displaying a set of lessons related to each other and to index terms.

browses through this library, and say he is interested in making up problems about Fortran format statements. He specifies to the corresponding problem generator that he would like a problem involving I, F, and E fields, for example, repetition factors, and nested parentheses. When his students take the exam, the generator will create an infinite variety of problem instances, all conforming to the general pattern specified by the instructor. Each student sees a different problem instance on his screen, each student can take the exam at his own time. Plato grades the problem interactively (if the student gets the first step of the solution wrong, the exam system corrects it right away, so he is not penalized during later steps). The instructor finds the exam graded the instant the student logs out. It's a great labor saving device.

KZ: What have been the shortcomings of Plato for your ACSES project?

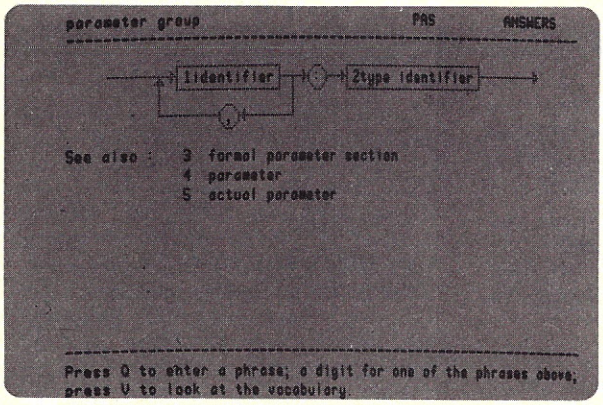
JN: Not many, I am very much impressed by Plato - it offers a collection of useful services not matched by any other system I know. But when you work with a system for a long time you always find something that could be improved. Primarily, the large size of a Plato system, and the high cost of a terminal and communications cost are major hindrances to its spreading into

Latest model PLATO plasma terminal couples excellent graphics with touch sensitive response. Control Data Corp. markets the PLATO system commercially.

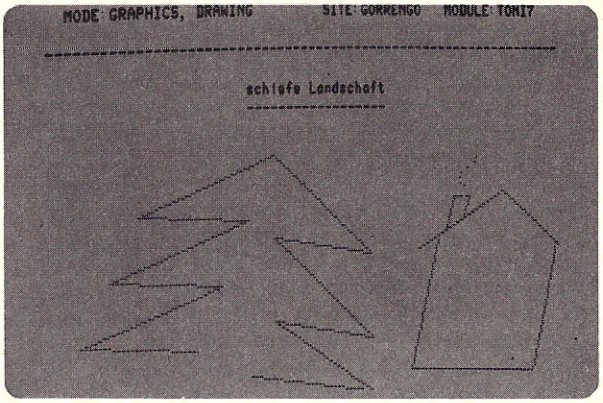
schools. If you can pack into a small stand-alone system the most desirable features of Plato, the market will be much greater. This is what I have tried to do with the system I talked about at the NAUCAL conference, the XS-0 school computer.

KZ: Are there other advantages a microcomputer offers besides low cost?

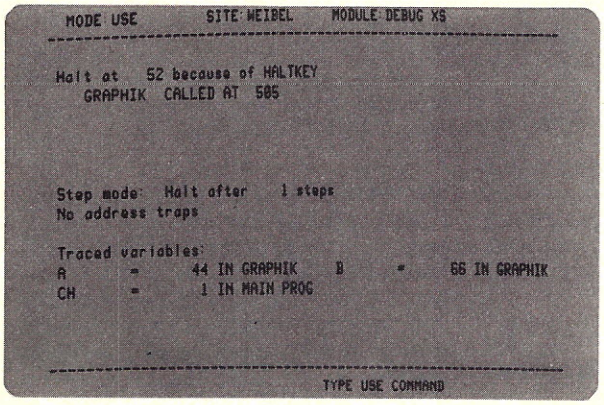
JN: A microcomputer in a terminal gives you faster response than a large time-shared system - on XS-0 we get instantaneous response (1/10-th of a second is perceived as instantaneous) most of the time. But the most impor-



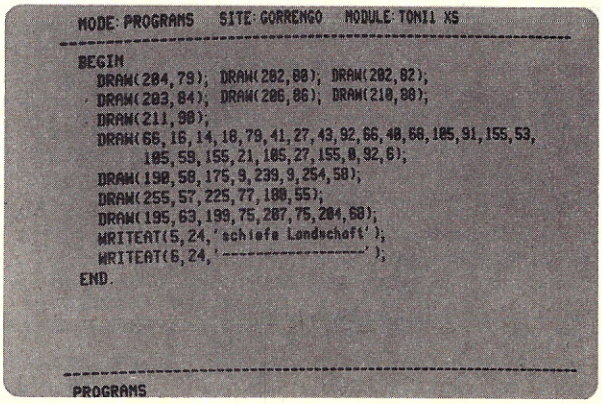
The self-explanatory school computer XS-0: An on-line manual displays syntax diagrams which define the user language PASCAL in response to a query.



The graphics editor allows a user to draw pictures on the screen, then generates automatically a program that displays this picture. This feature greatly facilitates writing illustrated instructional material.



The student can interrupt his program at any time during execution, and inquire about the current value of any variable.



tant goal is to make CAI systems much cheaper without giving up the good features that we have come to appreciate on Plato-animated graphics, and many high level aids for the author, for example graphics editors.

KZ: You have shown some slides taken off the screen of XS-0, but what is the essence of what you are trying to achieve?

JN: I want to develop the ideal school computer - a low cost interactive graphics system that serves as an automated desk for the student. It offers many services such as:
- text and picture editors
- good filing system to replace a pile of scraps of paper
- message system to communicate with other users
- interactive programming in a high level language, in our case a subset of Pascal
- instructional material on many subjects, including programming

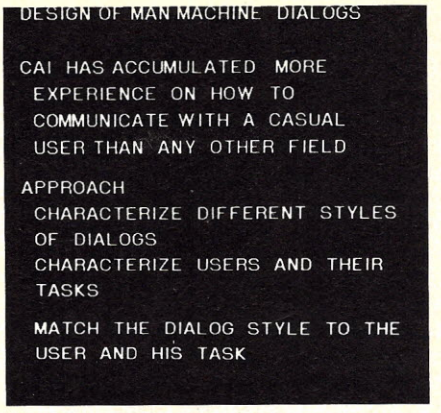
KZ: Does CAI have to be part of it, or would a sophisticated programming, editing and filing system be sufficient?

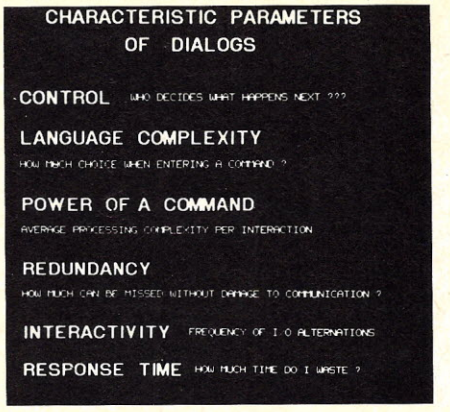
JN: CAI is becoming an integral part of all interactive systems. As more devices of all kinds incorporate computers, the possibility opens up to make these devices self-explanatory, that is, they provide instruction to the novice user on how these devices are operated. This is already happening with all kinds of desk top computers and smart terminals - for example, the Hewlett Packard 2644 and the IBM 5100 are delivered with instruction tapes.

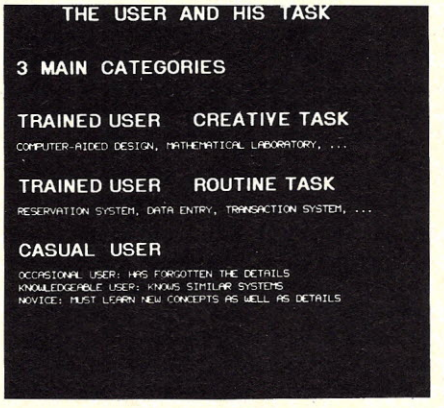
The quality of these kinds of instruction tapes is usually low compared to the best one has achieved in CAI - because the people who write them have no experience in writing courseware. In CAI we have accumulated more experience on how to write effective and informative and pleasant dialogs than in any other computer application area.

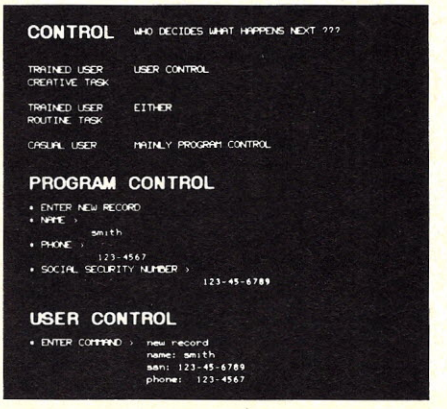
KZ: You have a great deal of experience teaching computer science and preparing curriculum materials. Perhaps you have some thoughts about the changes in computer science education, especially for non-professionals, that will come about because of low cost computing on interactive systems, automated program checking, and other aids made possible by a computer dedicated to one student.

JN: At many universities the first programming course is taught in huge sections - 200 students watching one professor wave his hands at the blackboard is a spectator sport. But programming is something one needs to learn by practice rather than by listening. The large section approach is very poor for teaching a practical skill like programming. Learning at a terminal and being able to switch at any time from the instructional mode to the laboratory mode to write and run programs is much more effective.









Guidelines for the design of effective man-machine dialogs.

KZ: I wonder if programming courses, and the teaching of programming on the job, will become common for professionals who might find computers useful?

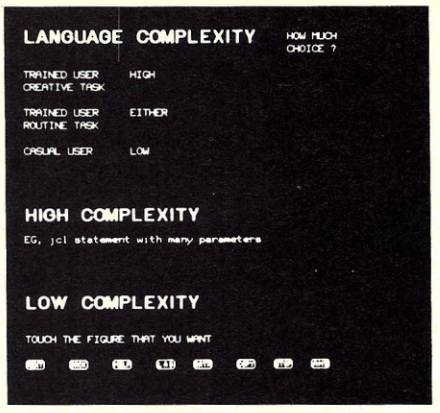
JN: Although lots of people will be programming in their daily work it may not look much like what we think of as programming today, say in Fortran. Special purpose, turn-key systems tailored to one application will account for the largest amount of computing. In a design application the user will draw pictures on a screen; in accounting he will enter parameters on a form; and so on. We call this automatic programming today, but whatever it is called, most people will use computers in this form, rather than programming them in procedure-oriented languages as we know them today. We have to widen our concept of what a programming language is.

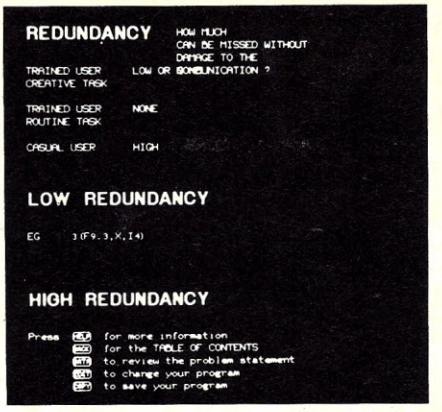
KZ: So computer scientists and educators ought to be doing something about that now?

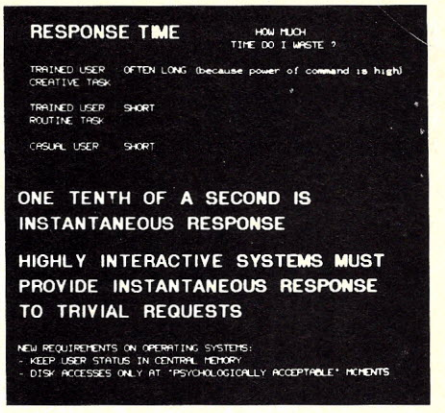
JN: Some do. Research on automatic programming is in that direction. So is research on man-machine communication. When you use a computer interactively, there is no sharp dividing line between selecting an item from a menu of options and programming in the conventional sense. There are languages that bridge the gap. One mini-language we implemented on Plato was designed for kids who cannot read or write yet. All the commands, all values of variables, everything is represented by pictures: balloons, motor cycles, trees, houses. The child can program animated cartoons in this language just by touching pictures, including symbols for start, move, stop. We have tried this with kids from 5 to 9 years and they easily program simple animation sequences, such as a boy getting on a motor cycle, riding across the screen, getting off and walking into a house.

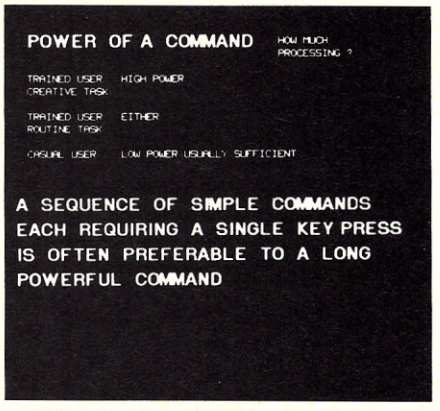
KZ: Does the kind of personal computing you have seen here at the NAUCAL 77 exhibit correspond to your ideas about interactive use of computers?

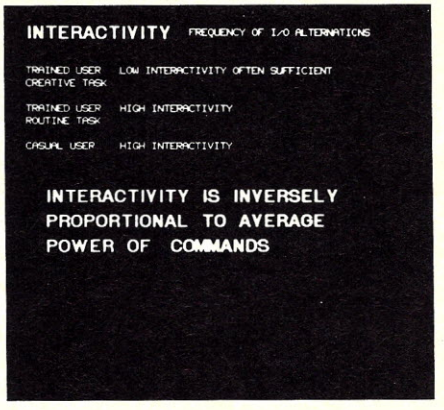
JN: Pretty much, since personal computing is by definition highly interactive. But many current systems don't take advantage of the fact that the user has unlimited access to the machine, and impose a style of man-machine dialog more appropriate to batch processing - for example, he types in a long command with many parameters, and the system provides feedback only after a lot of processing. There is no

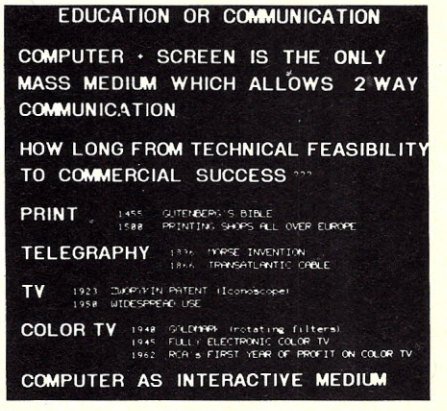












reason why a dedicated computer should not react to every single key press and interpret it as a command. At least this saves hitting RETURN all the time. We have a lot to learn about how to make computers work with a minimum of action on our part.

KZ: I will include several slides of your NAUCAL presentation because they summarize nicely your ideas on man-machine Communication. But why are we not further along?

JN: There is an art of designing man-machine dialogs which few people have learned today simply because we don't have enough experience. There are manuals on style for writing speeches, textbooks and the like, but none on how to write man-machine dialogs. In CAI much experience has been gained on how to write effective, informative dialogs. In a reservation or data entry system you pay the clerk for learning to interact with the system, but if a CAI dialog is poor the students will walk away from it.

KZ: Will you ever write that manual on man-machine dialog in education?

JN: I started 3 years ago to write a book on CAI but haven't gotten beyond the hundred pages I created then. Perhaps in a few years I will have time to continue.

KZ: What will you be doing in the next few years?

JN: I am still working on the low-cost interactive computer described in the NAUCAL presentation. It has taken all of my attention during the last two years, and I will spend another year making it a commercial product.

KZ: NAUCAL is one of the organizations which attracts speakers of national reputation to a relatively small conference of educators mostly from the region. People who don't get to ACM or NCC or IEEE meetings drive in each day to hear these national speakers and talk with each other. What is your reaction to NAUCAL 77?

JN: It's a friendly conference which provides many opportunities for conversation. The exhibits are interesting - the same computers in the $2000 range that are displayed at the personal computing exhibits. But I don't think these are going to be the school computers. The ones I see here are for the hobbyist to fiddle with, but teachers in schools will not have the time or skill to keep hobby computers running. And school computers should be able to offer more than Basic plus a lot of games.

KZ: The participation of micro vendors and computer stores at NAUCAL indicates they wish to appeal to the education market. Might they be wrong? Need the products shown here include a great deal more capability before they are viable in education?

JN: On the cost spectrum there is a gap at the moment. What is seen here is too small for a typical school. Most of the established mainframe manufacturers provide something too big and expensive. I designed XS-0 to fill this gap.

KZ: I look forward to seeing that product on the market before the end of 78! ■

The use of a personal computer: will be primarily for text processing (and, perhaps, for entertainment - playing games). Our tools for writing reports, letters, announcements, etc. are quite primitive. Right now I am editing copy of this interview on a PLATO terminal. Many things I have said here have been written up better in a number of papers I have written over the years. I could extend this particular write-up quickly by extracting and rearranging paragraphs from these earlier papers - if only I had them all stored on this system. But as things are, I have them on 50 scraps of paper, filed in many different places, and so I find myself writing pretty much the same things over and over again. I hope this waste of time will be incomprehensible to the generation of kids who grow up with computers today.

—JN

business computing

Inventory Control Overview:

The Micro-Princess and the Inventory Beast

Michael R. Levy

ITEM: 1978ADL
DESCRIPTION: WIDGET (LEFT-HANDED)
MAX: 100
MIN: 10
VENDOR:
location SHELF

			QUANTITIES				BALANCES		
			A	B	C	D	E	F	G
DATE	PURCHASE ORDER NO.	PRODUCTION ORDER NO.	ORDERED +"E" +"G"	RECEIVED +"E" −"F"	RESERVED −"O"	ISSUED −"F"	ON ORDER	ON HAND	AVAILABLE
1/2/78	3362	A100	100				100	0	100
1/10/78	3362	A100		100			0	100	100
1/15/78	3362	A100			20		0	100	80
1/20/78	3362	A100				20	0	80	80

Fig. 1. Written inventory card, used to check on how well a company controls its inventory.

With the discovery that the number of serious kit-building hobbyists is limited, and noticing the migration of the beginning enthusiast to the commercially-packaged "entertainment computer-in-a-box," the producers of complex microcomputer hardware are casting desperately about for new markets to insure survival. Along with the smaller commercial computer manufacturers, they consider the biggest growth area to be small business. Depending on which set of statistics you examine, this market segment includes ten to thirteen million eligible enterprises that produce 44% of the jobs, 36% of the GNP, and form 97% of all US business.

In their first attempts to approach this vast market, the microcomputer manufacturers have emulated their larger brethren and have quickly fastened on the "sacred seven" applications, as I call them: payroll, profit-and-loss statements, trial balance and general ledger, sales analysis, accounts payable, accounts receivable, and inventory.

The origin of these applications and the reason for their popularity date from the beginning of commercial computer applications. Large mainframe computers were applied to problems involving masses of paper and a very high volume of transactions. In big companies, this high volume of

transactions and paperwork was, and is, a logical target for computerization. Since "everybody has them," it was also natural for the computer companies to sell these applications. This tradition has continued with the introduction and popularization of classical minicomputers, and it seems to be the intent of the microcomputer manufacturers to emulate their older competition.

This imitative strategy may not adequately take into account the limitations imposed by the characteristics of small business and micro hardware and software. It is perhaps easier to examine this theory by using inventory as an example. Of the applications mentioned, inventory can be defined in more ways than the rest combined. It can mean anything from simple stockkeeping to the most complicated bill of materials and MRP (material requirements planning) schemes.

Let's examine simple stockkeeping and see what complexities develop. The arithmetic is deadly simple, but it must be kept straight. In our commercial work we use the accompanying card (Figure 1) to test our conception of the inventory procedures of the company with which we are working. We select an adequate number of sample parts to observe, and then we track those items and the company administrative procedures using the cards.

The card is divided into three vertical sections by two sets of double lines along with a top line containing some standard item description. The left-hand section contains date, purchase order, and internal job order or production order number for each data transaction. The middle vertical section is labeled *Quantities* and also contains the arithmetic instructions for the right-hand vertical section labeled *Balances*. For example, under column "A", line one, we order 100. The instruction above column "A" says + columns "E" and + "G", which we do. Next we receive the 100 widgets, and Column "B" says - "E" and + "F". For each line or transaction, the appropriate instructions are followed. The card then gives a good combination of usage, summary, and present-balance information, and it can then be used as the pattern for the design of a computer record.

We find that smaller companies, which have few administrative staff, cannot in some cases even keep track of a limited quantity of inventory items on the written sample cards, let alone on a computer. The key areas of trouble are usually receiving and the stockroom. Small-company personnel are accustomed to taking badly needed items directly from the receiving area to the production floor without benefit of receiving or stockroom paperwork. Stating the obvious, the balances will be inaccurate on a manual system or on a computer system if this happens.

It is one of my theories, which has now become almost an obsession, that the entrepreneur, who thrives and is successful in the fast-changing field of small business, works best from iterative intuitive judgment rather than from computer printouts. In most smaller companies, which would be the most likely targets for a hobby micro system, the transaction volume does not exist that would justify acquiring a computer for the "sacred seven" applications.

Besides the administrative problems, let's make a quick examination of the hardware and software required by our theoretical inventory system. We will assume that we have a well-disciplined company with a carefully policed stockroom and production floor, where everyone understands the necessity for accuracy. We will also assume an active inventory of 1000-plus items, all of which are changing usage and balances. Sequential files are difficult to use in such an environment, so that means that we have to have, at the very least, floppy-disk drives, and some sort of high-speed hard-copy device, which increases the cost of our system by a factor of two. If we are to update our file quickly and efficiently, we'll need triple drives. One diskette with the change transactions, one for the master file, and one to receive the new sorted and merged version of the master file derived from the floppies in first two drives.

Implicit in this arrangement is the use of several software system utilities such as file managers, and sorts. These provide standard methods for sorting and merging records, and methods for inserting and retrieving records for single-item status inquiry and for periodic reports. It is universally recognized by the commercial small-business system manufacturers that in order to be competitive, one must supply a full complement of system software aids and utilities. The hobby manufacturers, with rare exception, have avoided the whole software area. Instead they prefer to say that there are standard easily available packages — which is also a fable. In order to be "standard," the package must be written with a wide variety of options for use by many people. If it is written for use by a wide variety of small businesses it is, by definition, slow and inefficient for use by one specific business. Because of the uniqueness of most small businesses, even most "standard" commercial accounting packages end up being extensively customized in order to make them marginally satisfactory.

In any case, a small business contemplating the use of a micro should have developed some clear, documented, manual procedures that have been tried and operated successfully for an extensive period of time.

In the headlong rush into the accounting area, which collects *historical* information, most manufacturers have ignored the need for better *managerial* and *analytical* information. Earlier I said I felt most successful entrepreneurial types were highly intuitive. A small present-day micro system can provide significant aid to the business manager with the use of some simple iterative packages for bidding, forecasting, costing and the like. Whenever we discuss this type of integrated approach to small-business management, which we call LTBM - Little Toot Business Machine, in our seminars we are greeted with overwhelming enthusiasm by small-business owners. It's needed, so there appears to be a use for the micro in small business, but perhaps not in the areas which are specified by the present conventional wisdom. The inventory beast still hasn't been kissed by the micro-princess.

NOTES

If you want to know all about EOQ, MRP, forecasting, and demand indices in addition to simple stockkeeping, see the following more sexy jargon:

- Bourke, Richard W., *Bill of Materials the Key Building Block.* Pasadena, CA. Bourke & Associates. 1975.
- Wight, Oliver W., *Production and Inventory Management in the Computer Age.* Boston. Cahners Books. 1974.
- Plossl, G. W. and Wight, O. W., *Production and Inventory Control.* Englewood Cliffs, NJ. Prentice Hall. 1967.

For a good easily understandable discussion of why it's not a good idea to use sequential files in this kind of application see Chapter 1, the section on "Records and Files," in:

- Osborne, Adam, *An Introduction to Microcomputers, Volume 0 — The Beginners Book.* Berkeley, CA. Adam Osborne and Assoc., Inc. 1977.

[Ed. note: Jethro, founded by author Michael R. Levy, produces seminars in areas such as microcomputer business applications. The next ones will be given March 3-4 in San Jose, California, in conjunction with the Second West Coast Computer Faire.] ■

3rd Annual MACC Computerfest™ '78 June 23–25, 1978

Presented by the Midwest Affiliation of Computer Clubs

Featuring

- ★ Stupendous Hobbyist Exhibits
- ★ Tours and Evening Activities
- ★ Club Hospitality Suites
- ★ Special Club Meetings
- ★ Fabulous Programs
- ★ Manufactures Party
- ★ Technicial Sessions
- ★ Giant Flea Market
- ★ Free Seminars
- ★ New Products
- ★ Exhibits

Forget everthing you thought you knew about a computerfest. Because this is an all new computerfest in an all new city.

Quite simply, Detroit has what it takes to be the number 1 convention city in America and the MACC is holding the 1978 Computerfest™ in the Renaissance Center–a $500 million total environment complex one-third larger than Rockefeller Center. On 33 acres of landscaped riverfront. Come stay and relax in our new Detroit Plaza, which offers a convention environment unlike anything else on earth. With 70 stories of circular glass walls rising 800 feet, 1404 world class guest rooms, and special rates for our Computerfest™ guests. Secure, covered parking for thousands of cars. There is so much *action* happening during the three days of the fest, you'll want to *bring your walking shoes* when covering the over 100,000 square feet of convention facilities. We have provided for your complete comfort while at the Computerfest™ '78 with the following fantastic facilities *indoors* at the Renaissance Center:

- Eleven restaurants and lounges
- World's largest rooftop trilevel revolving lounge and restaurant complex
- Incredible walkways
- Outstanding recreation facilities
- Fantastic shops

There's even an eight story atrium with cocktail pods, hanging gardens and a giant reflecting pool where you can relax after those tough night programming sessions! Detroit's close to everyone with over 60% of America's population just one air hour away. The Detroit Plaza's located on the banks of the Detroit River. Seconds from Detroit's exotic Windsor tunnel. Within walking distance of adjacent Cobo Hall, Ford Theater, and the Civic Center. Foreign visitors are invited to fly into our ultra efficient International Metropolitan airport, just 30 minutes from our Computerfest™. For special motel/hotel rates and show reservation information, write: MACC Computerfest™, Box 9578 - Department LIT, Detroit, Michigan 48202 U.S.A. or call our 24 hour Computerfest hotline number (313)775-5320.

Make your *reservations NOW!*

CIRCLE 155 ON READER SERVICE CARD

business computing

Inventory Control:

Micros Unlimited (Computerland)

Ken Greene

One of the major necessities of any retail business, large or small, is the need to maintain accurate and timely information on the products it handles. Also very important is the ability to reduce the raw information on products and sales into meaningful reports on sales analyses to show profitability, product turnover, and many other such items. The "Inventory Control+" package written by Micros Unlimited was released last year and has proven to more than adequately provide this service in a very professional manner.

General Description

The inventory package is designed to operate on major fields which describe all information needed to maintain adequate records on a particular product. These fields are listed and explained in the table.

Item #
The item number the user assigns to an item. The range of item numbers is from 1 to 500.

Description
An 8-character alphanumeric field assigned by the user to briefly describe the item.

On order
Actual number of an item on order from a vendor.

QTY O/H
Designates the quantity of the item you have in stock.

QTY M-T-D
Number count of an item sold in a month.

Cost M-T-D
Actual cost to the user for the number of a particular item sold during a month. This item is updated as a function of sales.

Sales M-T-D
Dollar amount that the user actually receives during the month as a result of sales of the item.

Sales Y-T-D
Dollar amount the user actually receives during the year as a result of sales of the item.

QTY last year
Represents the amount (count) of the items sold during the past year. This is updated at yearly aging time.

Unit price
User's cost of an item.

Last sale
The last date when a particular item was sold.

KEY
A 10-digit alphanumeric code used to describe an item. The system has the flexibility to allow more than one item to have the same KEY.

Group
Designates the general category that an item falls under. This is a 10-character alphanumeric field.

Operation

The system is designed for the small-to-medium-sized business. Working with the program is rather straightforward. It consists of an executive Monitor called the "Load Monitor," and three application programs called INVEN, PRINT and UNUSED.

INVEN

The INVEN program is the main program of the package. Through this program, the user is able to perform the following functions:

Function Name	*Code #*
A. ADD	1
B. CHANGE	2
C. DELETE	3
D. AGE	4
E. REPORTS	5
F. MONITOR (RETURN TO)	0

Ken Greene, ComputerLand, 2 De Hart St., Morristown, NJ 07960.

A. *ADD* — This function adds a skeleton record to the inventory file, which will contain:

— Description. A field which describes the item. It is not used as a key for accessing data.
— Unit Price. Seller's cost per unit
— Key. Alphanumeric inventory key
— Group. Alphanumeric inventory group.

The video displays the following on calling INVEN.

```
ITEM #:
DESCRIPTION:
UNIT PRICE:
KEY:
GROUP:
```

Actual example:

```
SELECTION FUNCTION:   1
ITEM #:               25
DESCRIPTION:          HEX HEAD
KEY:                  10-150/4
GROUP:                BOLTS
```

The program has several operator defaults in case of an incorrect entry or previously assigned item. The program responds with "Input error — Retype" and "Item already exists," respectively.

B. *CHANGE* — This function allows for updating existing inventory records directly or as a result of order-update/sales processing.

An inventory item can be accessed by three different methods, Item #, Key, or Group. Upon access, all item information fields will be displayed as shown below:

```
KEY:                GROUP:
1- DESCRIPTION:     8- QTY Y-T-D
2- QTY O/H          9- COST Y-T-D
3- ON ORDER:       10- QTY LAST YEAR
4- QTY M-T-D:      11- UNIT PRICE:
5- COST M-T-D:     12- RECEIVED
6- SALES M-T-D:    13- NEW SALES
7- SALES Y-T-D:    14- LAST SALE:
                   15- NEW ORDERS
                   16- KEY/GROUP
```

Field to be Changed:

Using change allows very simple order-update processing by requiring the user to enter the code number of the field to be changed and enter the new data. Upon completion, the display of all fields will return with the updated records reflected.

C. *DELETE* — By entering the item number and description, all item information is cleared and the item may be reassigned.

D. *AGE* — The function clears the appropriate monthly or yearly *sales* information.

E. *REPORTS* — The reports entry allows the selection of three reports; sales profitability reports, item sales reports, and item stock reports. The reports can be selected by month or year and also by group or all (total) items in the inventory listings.

1. *Sales Profitability Report*

This report gives the main figures by month or year on sales, cost and profit of inventory items and prints grand totals. The report prints out under the following format

```
ITEM #   DESCRIPTION   COST M-T-D   SALES M-T-D   PROFIT/LOSS
```

2. *Item Sales Report*

The item sales report gives a monthly and yearly number count of sales of an item. The format prints out as follows:

```
ITEM #   DESCRIPTION   QTY M-T-D   QTY Y-T-D
```

3. *Item Stock Report*

The item stock report gives a count of the item in stock and the quantity on order. The printout is as follows:

```
ITEM #   DESCRIPTION   QTY O/H   QTY ON ORDER
```

PRINT Program

This program produces a cross-reference list of all items used under the following format.

```
ITEM #   DESCRIPTION   KEY   GROUP
```

UNUSED Program

This program prints out all unused ITEM #.

Hardware Requirement

The hardware needs for the system are 8080 or Z-80 based processor, 32K memory, and one (or more) North Star disk drives.

Summary

The Inventory Control + package meets and, in most cases, surpasses the needs of business for quality inventory control. The developers have added some extra goodies to aid users, like the ability to direct the reports output to a printer by merely answering Y to the prompt PRINTER (Y or N). There is a built-in copy routine to allow a single-drive user to easily duplicate disks. A final enhancement to the user is a newsletter to all registered owners of the package to give improvements and announce new systems as they become available.

AVAILABILITY

The Inventory Control + package is available directly from Micros Unlimited, Box 486, Stanhope, NJ 07874, and at most ComputerLand stores, for $500. ■

Mini/Micro 78

THE MINICOMPUTER/MICROCOMPUTER CONFERENCE AND EXPOSITION

APRIL 18–19–20 • CIVIC CENTER • PHILADELPHIA, PENNSYLVANIA

LOOK at over a half mile of product displays by the industry leaders! A "supermarket" of mini/micro computer systems, peripherals, and industry services all under one roof in the spacious Philadelphia Civic Center. The MINI/MICRO COMPUTER EXHIBITION will be the largest in the East in 1978, and one of the largest in the country. Many new products have been introduced at previous exhibitions.

LISTEN to almost 100 industry speakers in about 20 conference sessions discuss a wide array of topics of interest and value to you. And all sessions will be held in meeting rooms located just an escalator ride away from the exposition floor.

CONTRIBUTE your experience and expertise. A question, answer, and comment time is held at the end of each session at the MINI/MICRO COMPUTER CONFERENCE to encourage an informal and valuable interchange of information between both session speakers and those in the audience.

PROFIT by taking home facts and ideas important to you and the firm you represent. The opportunity to do so will be in Philadelphia this April 18–20. You'll be an important part of an estimated international audience of 10,000 industry professionals who'll look, listen, contribute, and profit by their attendance.

SPECIAL SEMINARS:

MONDAY, APRIL 17

MINI/MICROCOMPUTER APPLICATIONS

COURSE OBJECTIVE: Beginning with a brief review of microcomputer hardware and software, this applications course is intended to build on your knowledge of basic hardware configurations, memory systems, I/O Schema, and debugging methods. Understanding the differences in approach for applying minicomputers and microcomputers will be the theme of the course. The emphasis will be on microcomputer applications. Specifically, the software development process, development of the hardware system, hardware/software tradeoffs, interfacing, system specification, and some development cases will be covered. A general understanding of the process is one goal of the course. The course will close with an explanation of the important highlights of the hardware development process.

COURSE OUTLINE:

1. Reminder on current minicomputer characteristics and capabilities.
2. Review of microcomputer hardware and software.
3. The software development process.
4. Development of the hardware system.
5. Hardware, software tradeoffs.
6. Interfacing.
7. System Specification.
8. Some Development Case Studies.

Sponsor: The Institute of Electrical and Electronics Engineers (IEEE)

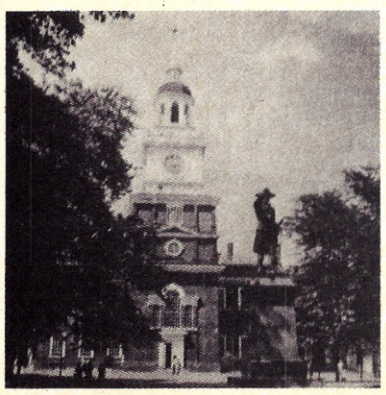
WEDNESDAY, APRIL 19

STEP-BY-STEP DESIGN OF MICROPROCESSOR SYSTEMS

The aim of the course is to expose the participants to step-by-step procedures for the design and implementation of microprocessor systems using the following modes of operation: (1) Wait/go; (2) Test-and-go (test and skip); (3) Interrupts; and (4) Direct Memory Access.

The design procedures which are accomplished in five well-defined steps, will be demonstrated and verified experimentally in class. Lecturer: Prof. D. Zissos, The University of Calgary, Canada.

Sponsor: The International Society for Mini and Microcomputers (ISMM)



To: **MINI/MICRO COMPUTER CONFERENCE AND EXPOSITION**
5528 E. La Palma Avenue, Suite I, Anaheim, CA 92807, Phone: (714) 528-2400

My Primary Interest Is:

☐ **Attending.** Please send me a Preview Program listing information on sessions/papers, exhibitors, and hotel reservations.

☐ **Exhibiting.** Please send a copy of the Exhibit Prospectus.

Please type or print

Name_____

Title_____

Company_____

Address_____ M/S_____

City_____ State_____ Zip_____

business computing

Inventory Control:

PolyMorphic Systems

Don Williams

PolyMorphic System 8813, with triple disk drive, video monitor, and keyboard, as used with the inventory system.

This article describes a solution to a problem faced by most businessmen: how to better manage a major asset of their business—the stock of merchandise they sell or use in their business.

My job at PolyMorphic Systems is to visit our dealers, usually spending the day with them and their sales staff. I demonstrate our equipment, discuss sales and demonstration techniques, and usually leave some demonstration programs with them. The first store I visited after the Atlantic City Personal Computing Show with the System 8810 we demonstrated at the show was the General Computer Company store in Detroit. That afternoon, two retail customers, never having seen a "personal computer," came into the store. Both, one a jeweler and the other the proprietor of several stereo stores, had the same question, "Can it do inventory?"

Of course the answer was "Yes." But both left the store somewhat disappointed that we could not show them the 8810 or any other computer doing much more than a few simple games.

The series of programs described here was designed to remedy that problem, as well as to make any demonstration believable by giving our dealers a system on which they could keep their own store inventory.

Don Williams, PolyMorphic Systems, 460 Ward Dr., Santa Barbara, CA 93111.

Description of Data Field	*Number of Characters*
Item Number	6
Item Class	1
Vendor Number Code	3
Item Description	23
Balance on Hand	5
Minimum Balance	3
Average Cost	8
Selling Price	8
Date of Last Issue	4
Date of Last Receipt	4
Issues to Date	6
Outstanding Purchase Orders	
First P.O. Number	4
First P.O. Quantity	4
First P.O. Date	4
Second P.O. Number	4
Second P.O. Quantity	4
Second P.O. Date	4
Third P.O. Number	4
Third P.O. Number	4
Third P.O. Date	4

Fig. 1. Information contained in each inventory record.

Problems to be Solved

Before describing the system and what it does, let's look briefly at what problems an inventory management system must solve. Most businessmen, particularly retailers, have a common problem of managing a major asset, which is the stock of merchandise they sell. They must be able to determine how much to stock of each item they sell or use so they can maximize sales and minimize their capital investment. They must be able to answer questions such as these:

How many gidgets do I have on hand?
How many widgets do I have on order?
Is the total I have "on hand/on order" enough to meet my expected sales?
When should I place an order for more gadgets?
What's the total value of my investment in widgets?
What's the total value of all the items I keep in stock?

With even the smallest number of items on which records must be kept, the clerical tasks of correlating the data about sales, receipts, costs, usage, and outstanding purchase orders is complex and time-consuming. Because of the complexity, most businessmen must go to several sources to answer these questions.

All this is accomplished by...merely putting the diskette in the disk drive and turning the power on or hitting the load button....

This system has been designed to assist the small businessman in managing his inventory by providing a means of recording information about each transaction that affects his inventory and assimilating this information into a single record for all items kept in stock. This file of records will then allow the businessman to have a single source for the information necessary to more effectively manage and control his investment in inventory.

System Programs

Three major programs make up the system:

1. INVENTORY maintains a file of records, one record for each item

INVENTORY TRANSACTION REGISTER 12/11/77 (BELOW MINIMUM =***)

ITEM #	DESCRIPTION	TRANS CODE	TRANSACTION QUANTITY	NEW BALANCE	
9015	Volley Ball-Leather	ADD ITEM	0	0	
P.O. # 4365 DATE= 1209				15	
9015	Volley Ball-Leather	RECEIPT	10	10	
P.O. # 4365 DATE= 1209				5	
9015	Volley Ball-Leather	ISSUE	3	7	
P.O. # 4365 DATE= 1209				5	
9015	Volley Ball-Leather	ISSUE	7	0	***
P.O. # 4365 DATE= 1209				5	
9015	Volley Ball-Leather	P. O.	0	0	***
NEW SELLING PRICE= 12.95					
9015	Volley Ball-Leather	ADJUSTMENT	0	0	***
NEW VENDOR CODE= 35					
9015	Volley Ball-Leather	ADJUSTMENT	0	0	***
9015	Volley Ball-Leather	DEL ITEM	0	0	***
9015	NOT USED----------	DEL ITEM	0	0	***

Fig. 2. Printout of Transaction Register, which maintains a file of one record for each item stocked.

STOCK STATUS REPORT December 12, 1977 (BELOW MINIMUM=***)

ITEM #	DESCRIPTION	SALES PRICE	AVERAGE COST	USE T/D	BALANCE		INVENTORY VALUE
9001	BASEBALL BAT	9.95	6.98	113	34		237.32
	P.O. # 3703 DATE: 1201				5		
	P.O. # 4561 DATE: 1209				25		
9002	FIELDER'S GLOVE-LEATHER	27.95	16.50	12	7		115.50
9003	CATCHER'S MIT-LEATHER	34.95	21.50	5	2	***	43.00
9004	1st BASE MITT-LEATHER	29.95	18.00	7	3		54.00
9005	BASEBALL SHOES-SZ 8	22.95	14.75	6	3		44.25
9006	SOCCER SHOES-SZ 8	19.95	12.50	5	5		62.50
	P.O. # 3602 DATE: 1115				10		
9007	SOCCER BALL-LEATHER	12.95	7.50	8	12		90.00
	P.O. # 3609 DATE: 1015				5		
9008	BASKETBALL-LEATHER	14.95	9.05	8	7		63.35
	P.O. # 5001 DATE: 1015				15		
9009	VIDEO GAME-4	29.95	17.50	0	0		.00
	P.O. # 3501 DATE: 1121				50		
9010	SNOW SHOES	49.95	24.50	0	0	***	.00
			TOTAL VALUE CLASS 9 IS				$709.92

Fig. 3. Printout of Stock Status Report, listing all items in the file, showing quantity on hand, on order, and value.

stocked. The information kept in the file for each item is listed in Figure 1. Optionally, a *Transaction Register* may be printed to be used primarily as an "audit trail." Figure 2 is a sample of the Transaction Register.

2. STOCK-STATUS prints a *Stock Status Report.* This is a printed list of all items in the file showing the current quantity on hand, on order, and value. Figure 3 is a sample of the Stock Status Report.

3. STOCK "prints" a "soft copy" of the Stock Status Report on the video screen.

The INVENTORY program contains seven major functional capabilities. It can:

1. Inquire as to the status of any item that is stocked.
2. Post receipts to the inventory records.
3. Post issues/sales to the inventory records.
4. Record up to three outstanding purchase orders on any inventory item's record.

The program checks the information entered by the operator for validity and completeness....

5. Create records in the inventory file for new items that are to be stocked.
6. Delete the records of inventory items that are no longer to be stocked.
7. Adjust prices, costs, balances, and minimum balances for any item kept in stock.

Disk Operating System

Using the "initial" capability of the System 8813's Disk Operating System (DOS), when a diskette with the INVENTORY program is placed into Drive One of the system, DOS looks at the disk directory and determines that there is a file named INITIAL.BS (which we have named our INVENTORY program. DOS determines that INITIAL is a BASIC source program, loads BASIC, loads the file name INITIAL.BS, and then executes the inventory program. All of this is accomplished by the operator merely putting the diskette in the disk drive and either turning the power on or hitting the load button on the front of the System 8813.

```
                    INVENTORY CONTROL
Please make SELECTION from the following list:

     1.  INQUIRY
     2.  RECEIPT
     3.  ISSUE/SALE
     4.  PURCHASE ORDER-Add or Delete
     5.  ADD NEW ITEM TO FILE
     6.  DELETE ITEM FROM FILE
     7.  ADJUSTMENT-Balance,Cost, Price, Reorder Point
                    Vendor Number,Issues-to-date
     8.  STOP

Enter NUMBER of Selection  : 1
Item Number (xxxx)         : 9002
```

Fig. 4. The MENU shows all the functions that INVENTORY can perform. The operator has requested inventory status, by entering a 1.

```
                    ADD NEW ITEM
Item Number              :  9017  Class:  9
Description              :  SKI POLES
Vendor Number            :  56
P.O. #1 (nnnn):5123  Quantity (nnn):6  P.O. Date (mmdd):1215
P.O. #2 (nnnn):0
Date of Last Receipt    (mmdd):  0000
Date of Last Issue/Sale (mmdd):  0000
List Price           (nnnnn.nn):  49.95
Average Cost         (nnnnn.nn):  32.50
Quantity On Hand        (nnnnn):  0
Re-order Point            (nnn):  2
Issues-to-date          (nnnnnn):  0
.............is DATA OK.....................? Y
```

Fig. 5. Information displayed when adding a new item to inventory, even though, in this case, none are on hand. But they're coming.

```
FIELDER'S GLOVE-LEATHER      Item Number:  9002  Vendor: 2
P.O. Number          P.O. Date          Quantity
 4892                 1123               15
 0                    0                  0
 0                    0                  0
Date Last Receipt :  609   Date Last Issue:  915
Average Cost    :      $16.50    Inventory Value:      $115.50
Selling Price   :      $27.95    Issues-to-date :      12
Balance On Hand:  7              On order       :      15
Re-order Point :  5

Please enter QUANTITY received     : 10
Please enter UNIT COST
...hit RETURN if NO CHANGE.........:
Please enter PURCHASE ORDER NUMBER : 4892
Please enter DATE RECEIVED         : 
```

Fig. 6. Same item as in Fig. 5, but now the computer is recording receipt of ten of the fielder's gloves.

```
FIELDER'S GLOVE-LEATHER      Item Number:  9002  Vendor: 2
P.O. Number          P.O. Date          Quantity
 4892                 1123               5
 0                    0                  0
 0                    0                  0
Date Last Receipt :  1215   Date Last Issue:  1215
Average Cost    :      $16.50    Inventory Value:      $198.00
Selling Price   :      $27.95    Issues-to-date :      17
Balance On Hand:  12             On order       :      5
Re-order Point :  5

1.  COST     2.  SELLING PRICE   3.  BALANCE
4.  REORDER POINT   5. VENDOR NUMBER  6.ISSUES-TO-DATE
Which ITEM do you want to change (1,2,3,4,5,or 6) : 1

Enter new UNIT COST : 17.95
```

Fig. 7. Now an adjustment is being made on the same item, because the unit cost has been raised.

After the operator enters today's date, a MENU of the functional capabilities that INVENTORY can perform appears on the video screen.

The person operating the System 8813 selects a number according to the type of *transaction* they want processed. Figure 4 illustrates the video display of the MENU after the operator has entered a 1 for the SELECTION and a 9002 for the Item Number, indicating that they would like to inquire as to the inventory status of Item Number 9002.

Figures 5, 6 and 7 illustrate the information displayed when adding new items, processing receipts and making adjustments.

Interactive Mode

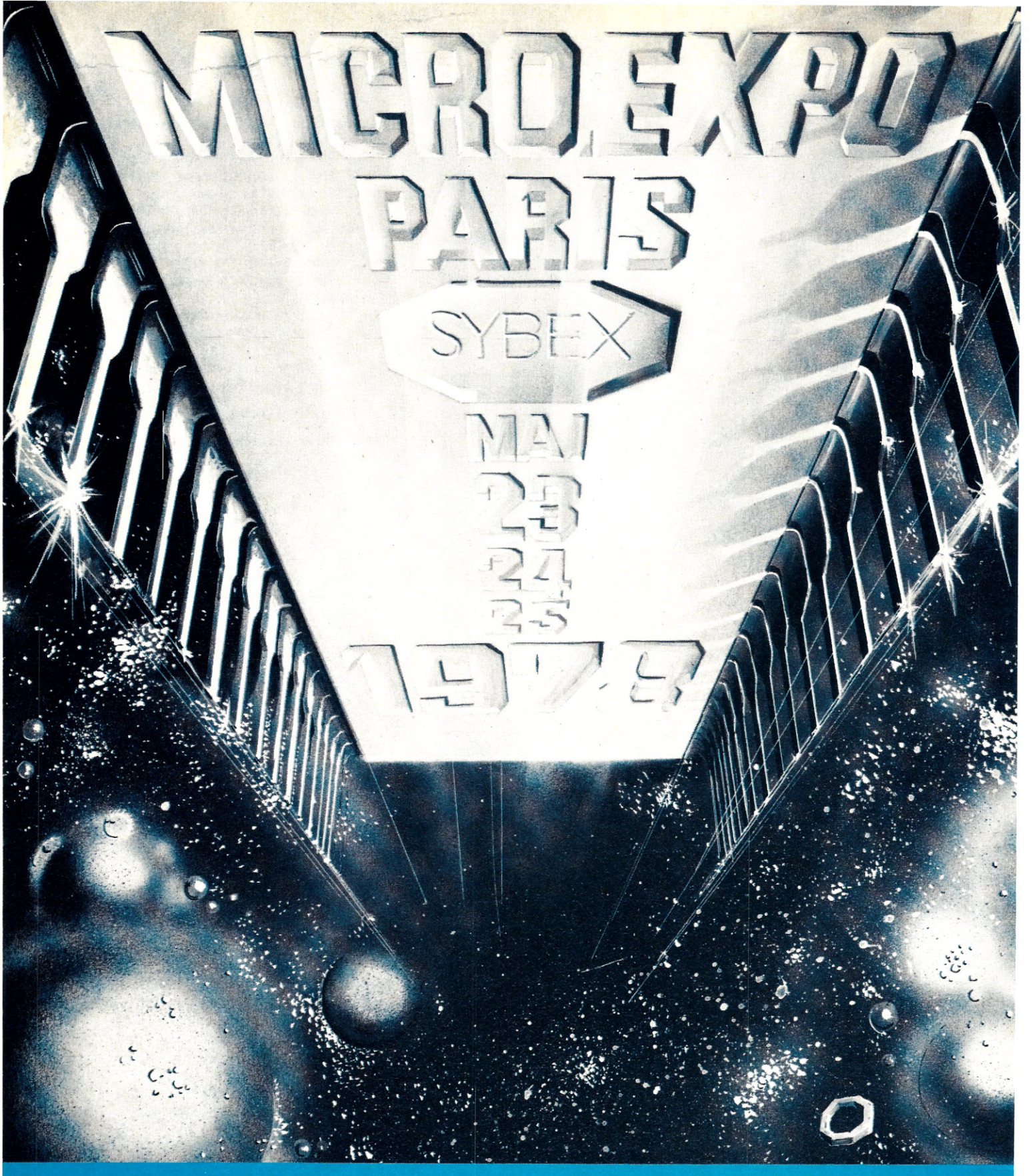
This INVENTORY program operates in a completely *interactive* mode. That is, after displaying some information, the system prompts the operator for each piece of information needed to process a transaction. The 8813 then immediately processes the information by finding the appropriate item's record in the file on the diskette, reads the record into memory, updates the record, and then writes the updated record back to the disk file.

This is in contrast to the more classical *batch* processing method where all of the transactions affecting inventory are batched up, transcribed to another media, arranged by item, and then posted by the computer to their appropriate records. The significance of this system being *interactive* rather than *batch* is twofold. The program checks the information entered by the operator for validity and completeness as it is entered. The operator will normally be someone knowledgeable about the inventory. These two factors will normally increase the accuracy of the inventory records significantly, compared to a batch processing system.

Dedicated Computer

Secondly, because of the relatively low cost of the System 8813 and similar microcomputers, it is practical to *dedicate* a system to this one job. Therefore, as inventory activity takes place, this activity can be recorded in the file. Thus the inventory records can be more current and useful.

This system will operate on PolyMorphic's System 8813 with two disk drives and 24K of memory. A single diskette can contain the records for over 900 inventory items. All of the programs were written entirely in BASIC. Any PolyMorphic Systems dealer with a System 8813 demonstrator can demonstrate the system. ■



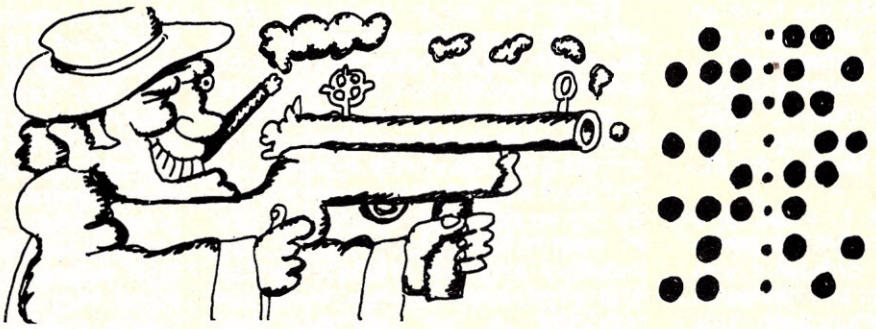
INFORMATION - INVITATIONS:

EUROPE:
313 rue Lecourbe
75015 - PARIS
Tel: (1) 828-2502

USA:
2161 Shattuck Ave.
Berkeley, Calif 94704
Tel: (415) 848-8233

business computing

Inventory Control:
AIM (Computer Mart of NJ)



Tom Cirillo

The design and implementation of the AIM System (Automatic Inventory Management System) was motivated by an actual requirement for inventory management on the part of the Computer Mart of New Jersey. At first, it seemed clear, simply on the basis of the high level of sales orders and purchase orders, that automating the inventory function would solve two problems by providing: (1) an accurate picture of the exact inventory status could be had at any time along with sufficient information about sales order and purchase order activity, and (2) the foundation for subsequent automation of accounting functions. As a result, an initial study was made of the requirements of the Computer Mart of New Jersey for an inventory management system. As is the case with many system studies of this type, it became apparent that relatively small increases in development effort would result in a system that was far more sophisticated than originally requested, and at the same time, would be sufficiently generalized to be able to be offered in the marketplace. This study also included a brief analysis of the then available inventory packages. The conclusion reached at that time was that none of the available systems could satisfy all of our requirements due to either the simplicity of their system design, or what appeared to be a lack of comprehensive business understanding of the functions of an inventory system.

Tom Cirillo, Computer Mart of New Jersey, 501 Route 27, Iselin, NJ 08830.

Functions Included

The result of the study was the determination to develop a generalized inventory management system that would be usable by a variety of different businesses whose requirements might vary from ours. Simply stated, of course, the functions to be included in this design are not extraordinary in any particular sense. They include:

- complete inventory maintenance
- sales order entry subsystem
- purchase order entry subsystem
- provision for capturing sales history
- automatic handling of back orders.

Since these functions are typically a part of almost every inventory system, this article will illustrate the essential differences in our approach that resulted in a much more functional system in each of these areas.

Hardware Components

After having conceived the system design, and having given serious thought to the performance characteristics required of the system, a final selection of hardware components was made. Since this article does not concern itself with the criteria for selecting hardware, I will dispense with the particulars of our selection process except to say that we were fascinated with the Micropolis disk-drive system and intended to utilize it in our system configuration. The resulting configuration is:

- IMSAI CPU with either 32 or 40K of memory
- Lear Seigler ADM-3a CRT
- single or dual-drive Micropolis Disk System (with Micropolis Basic)
- printer optional
- all appropriate interfaces, etc.

Naturally, this hardware configuration can vary by component within the confines of architectural compatibility.

One of the pitfalls of other inventory systems we had examined was their lack of capacity (that is, they would not handle a reasonable number of inventory items and purchase-order items). This was one reason for our fascination with the Micropolis disk-drive. Its cost/capacity ratio is impressive and we hoped to utilize this to solve the problems that became apparent in the initial phases of our system design. For example, as a result of our design approach, we are able to handle well over three thousand inventory items on a single diskette. Furthermore, we can also accommodate almost five thousand purchase order line items per diskette. Of even greater significance,

our purchase order items can be entered in any order and can be accessed randomly. This relies heavily on the capacity and format of the Micropolis disk-drive.

Overview of System Design

Before describing in detail the file structure and access techniques we designed, and processing implications they entail, let's overview the system design and see exactly how data is processed by the AIM System.

The entire AIM System is "menu-driven." This means that the selection of a particular system function simply entails finding it on a menu which appears on the CRT, and keying in the number associated with the function you wish to perform. The result of this selection will be the invocation of the appropriate program which will perform the desired function. Upon completion of processing, this program will return control to the master menu.

Referring to Figure 1, which is a description of the entire AIM System in terms of the major programs and master files, we can get an overview of all of the AIM System functions. This diagram indicates that there are five major files within the AIM System:

1. Vendor File (VENDOR)
2. Inventory Master File (INVMAST3)
3. Purchase Order File (POINDEX & PODATA)
4. Sales History File (SALESHIST)
5. Backorder File (BOFILE)

These five files represent the storage means within which all data is handled in the AIM System. Further inspection of Figure 1 indicates that there are also five major programs in the AIM System:

1. Vendor File Maintenance Program (VENDMAINT)
2. Inventory Master File Maintenance Program (ACCINV3)
3. Purchase Order File Maintenance Program (POMAINT)
4. Sales Order Entry Program (SALESORDER)
5. Back Order Program (RELBO)

Specific Inventory Functions

These five major programs (and other utility programs) perform all of the major data-processing functions within the system. Now, let's take a look at the specific functions one wants to perform in an inventory system. Using our previous list of requirements, we first dealt with inventory maintenance. By this we mean the ability to maintain all information associated with any inventory item our business carries. This information includes, but is not restricted to, an inventory item number, an item description, a vendor number, the price, quantities, minimum order levels, etc. This information and more is captured in inventory master file records in INVMAST3. Before discussing the particular manner in which the records are maintained in the file, let's first introduce the concept of "logical records."

Since one of our original system requirements was that the system have a reasonable capacity (for an average small business), one of our design goals was to maximize use of the already dense Micropolis disk format. Since the unstructured format involves reading and writing blocks of data 250 bytes long, if we were to assign one inventory record (or for that matter, any type of record in the system) to each block, our diskette capacity would be restricted by the number of sectors (in this case, approximately 1200). Besides the fact that 1200 inventory items was deemed to be insufficient, even at this level, the entire diskette would be dedicated to the inventory master file, also an unreasonable restriction. As a result, we chose to utilize the concept of "logical records" in various areas of the system. Inventory-item logical records are simply as long as the number of bytes required to fully describe an inventory item (regardless of sector size). We found that we could define 66-byte logical records, giving us room for three records per sector. This results in the ability to store over 3600 inventory records on a single diskette. Other files within the system have different "blocking factors" (the number of logical records per sector).

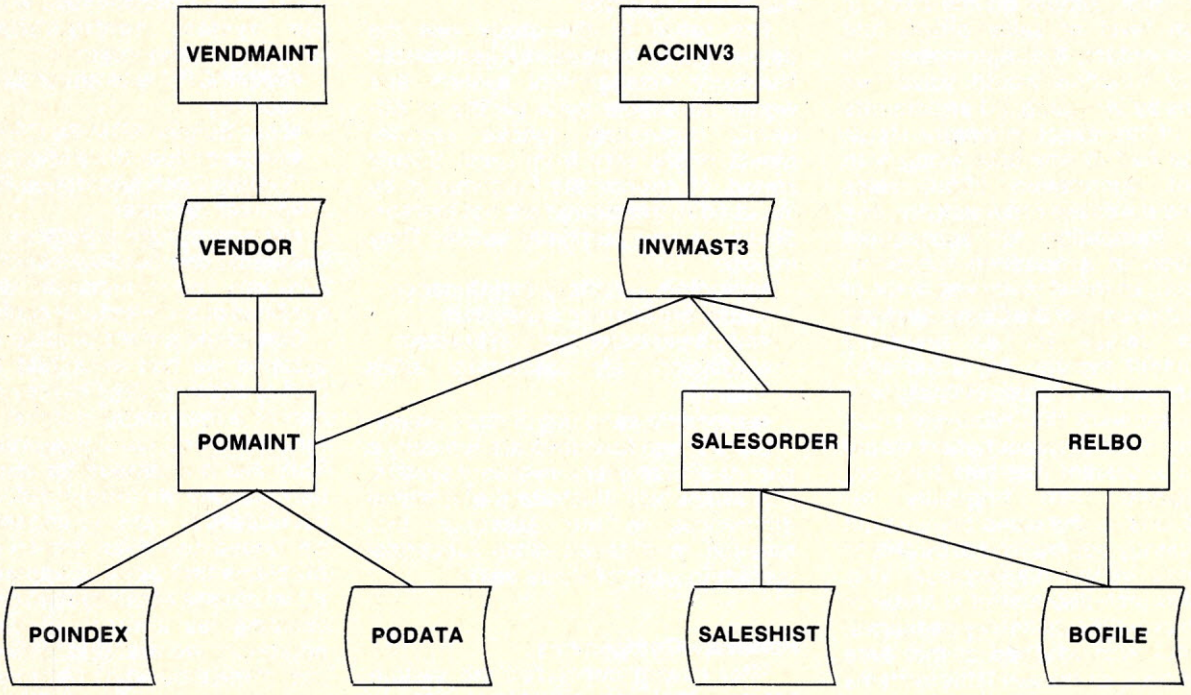


Fig. 1. Entire AIM system, showing major programs and master files.

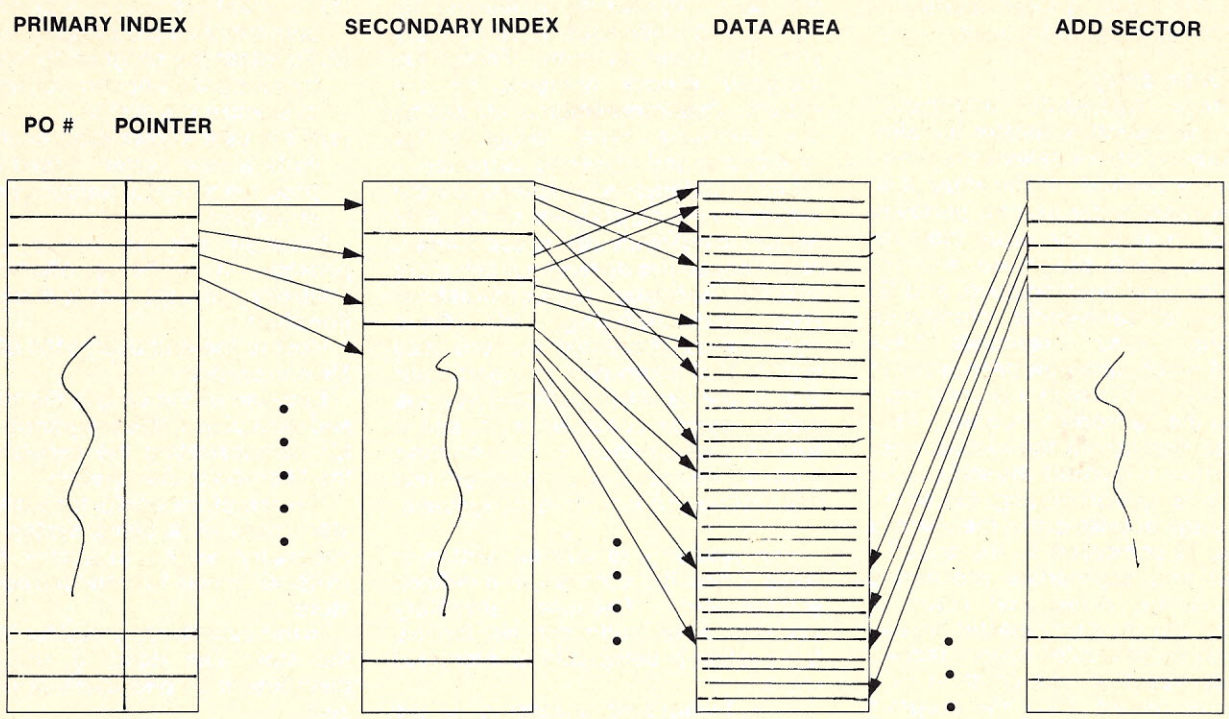


Fig. 2. Organization of the AIM indexed file.

Getting back to our inventory master file, it is apparent from Figure 1 that many functions within the system affect this file. In particular, the inventory master-file maintenance program provides the ability to change any of the fields within the inventory file as well as create new inventory records as new items are carried by the business. Furthermore, as indicated by the connecting lines, the Purchase Order Maintenance program, the Sales Order program, and the Back Order program all affect the inventory master file. Let me now describe these functions and the effect they have on this file and others.

Purchase Order Maintenance

The Purchase Order Maintenance program enables a user to create a new purchase order, make an inquiry as to the status of a purchase order, or receive against particular PO line items. Once the program is selected, a menu appears allowing further selection of each of these functions. When you choose to add a purchase order, you must enter all of the fixed header information required by the program (that is, all the fixed information describing this particular purchase order such as vendor number, etc.). The purchase-order number may be any number and is not bound by chronology (the significance of this will be described later). After entering a header record, you must enter at least one line item and may enter as many as nine line items to fill out your entire purchase order. If you wish to inquire and/or receive against a purchase order, you simply select the purchase order by entering its PO number and the screen will fill up with the header and all line items associated with it. After you have all of this information, you may "receive against" any specific line item. If you choose to do so, the appropriate PO line-item fields will be modified on the screen and in the file, and will result in the updating of the inventory master file as well. When you have completed receiving against any or all of the PO line items, control will be returned to the original purchase order menu.

I previously mentioned that this program does not require chronological entry of purchase orders when they are created. This means that if your business uses pre-printed purchase-order forms and a number of individuals have responsibility for issuing a PO, there is no danger of either an out-of-sequence condition arising, or a PO being prevented from being entered into the system; PO numbers can be issued at random as far as the system is concerned. This was a critical requirement in the system design and was accomplished through the use of an indexed file. Figure 2 illustrates the organization of our indexed file technique. As you can see, when looking for a purchase-order record, the program first searches a primary index to determine the sector in which the appropriate pointer to the data area can be found. This set of sectors is called the secondary index. Records are randomly stored in the data area. However, the secondary index contains the PO number (the "key") of each record in the data area, sequentially filed. The primary index, as indicated by the arrows, contains the first PO number to be found in each of the sectors of the secondary index. It too, therefore, is sequential. As a result, this technique is generally known as indexed sequential access. Note also that there is an "add" sector. This add sector contains the PO number and relative record number of each new purchase order added to the file. When this add sector becomes full, a reorganization of the purchase-order file is performed. The reorganization program automatically re-sorts and re-establishes the pointers between the various indices within the file. The process is highly efficient and can be run on demand. The result of this technique is an extremely powerful access method which enables the user to randomly — that is, directly — access purchase orders which would

other wise be accessed in a highly rigid and cumbersome fashion. Furthermore, the actual access method is totally transparent to the user.

Sales Order Entry

Naturally, sales-order information affects the inventory master file also. The purpose of our sales-order entry program is twofold: (1) the entry of an original sales order and all pertinent information about the order after the order has been processed and the quantities have been shipped, and (2) the entry of sales-order information regarding items previously back ordered which have now been satisfied (notification of which resulted from running the backorder program). As is the case throughout the system, after the program is invoked, the sales-order program is self-prompting. On entering an original sales order, the quantity shipped is compared to the quantity ordered and appropriate processing decisions are made that result in updating the inventory master file and the creation of a sales history record. This can result in the creation of a backorder record. Also, the program will respond to information regarding the willingness of a customer to accept a partial shipment. If the sales-order information being entered represents that of an item previously entered which was backordered, the result is slightly different. The inventory master file is updated to reflect the satisfaction of the back order. Naturally, in all cases, all appropriate sales-history records are created.

Back Orders

In the course of adding sales orders to the system, there will be an inevitable buildup of back orders. From time to time, we wish to relieve the back order file of these records when they can be satisfied (that is, goods have been received). Various techniques to release back orders can be utilized by an inventory system and, as a result, the AIM System is programmed to allow a user to modularly implement any special technique. Our user required the release of back orders of a FIFO (first in, first out) basis. Figure 1 indicates that the Back Order Release Program, RELBO, acts against the back order file and inventory master file. This program determines whether the inventory master file "available" quantity is sufficient to satisfy a given back order. RELBO sequentially processes the entire back order file and handles both full and partial shipment situations. The result of this processing step is the release of all back orders that can be satisfied, the production of a report indicating this, and the appropriate updating of the inventory master file.

Referring back to Figure 1, you'll note that the AIM System also contains a vendor maintenance program, which maintains a vendor file containing the names and addresses of all vendors you do business with. These are assigned vendor numbers by the system. This information is utilized by the purchase order program for verification and reporting purposes.

I have endeavored to summarize the features and capabilities of the AIM System in relatively little space. Certain of its capabilities deserve a much more detailed description for a comprehensive understanding. We have nevertheless presented the business factors that motivated this particular design, the hardware selected, various system design considerations, and a summary description of the resulting product, including the characteristics that distinguish it from other systems.

The system was initially written to solve a specific problem in a defined environment. Because inventory systems have sufficient similarities, this system is being sold to interested users.

It is being sold according to the following formula:

End User

This applies to a one-time end user for a particular application. We recognize that no system, even the most sophisticated, will meet the needs of all users. So if the structure of this program is appealing to an end user he has three choices:

(a) To adapt his business practices to conform with the system.

(b) To adapt the program to his needs by hiring a computer consultant to customize the system.

(c) If the user knows programming, to write a new system based on this program's logical functions and file structures.

Because this is not a turnkey package, it is being offered at a relatively low cost with a manual and flowchart.

Cost to the end user, $100.00.

Demonstration

Because of the unique file structure and ease of use, this program is superb for demonstrating the capabilities of the Micropolis disk system.

It is therefore, useful for a computer store both as a demo system and if necessary as an inventory system. (After all, it was written for a computer store).

In this case it is licensed for $500 and the store can demo it, use it and distribute it to end users at no extra cost.

Turnkey

We can customize the system to your needs on a fixed-fee basis. If this system is close to what you've been looking for, we could customize it at a far more reasonable rate than writing an inventory system from scratch. ■

Let's Get Personal in Anaheim

June 6-8, 1978

A rewarding personal experience is in store for you June 6-8 at the NCC '78 Personal Computing Festival... the most comprehensive personal computing event ever held. The Festival, a separate feature of the National Computer Conference, will include approximately 30 program sessions, commercial exhibits of consumer computing products and services, plus a contest and exhibit of microprocessor systems and applications. All Festival activities will take place in the Disneyland Hotel Complex, just a few minutes from the Anaheim Convention Center, site of this year's NCC.

Plan now to attend the big, new NCC '78 Personal Computing Festival. The program will include special paper, panel, and tutorial sessions on such topics as speech synthesis and recognition, computerized music systems, hardware and software design, computer graphics, and small business systems. All papers will be published in a softbound volume, *Festival Digest '78*, which will be available during NCC.

Festival exhibits will provide an extensive display of commercial offerings by organizations serving the personal computing field. More than 100 companies, occupying over 175 booths, will display systems, components, terminals, software, kits, disc and tape cassettes, relevant publications, and related hobby items.

Rounding out the Festival will be a contest featuring microprocessor systems, devices, and applications ranging from home-brew DOS and graphics terminals to educational applications and computer games. Prizes will be awarded for the best exhibits.

Don't miss the year's most exciting personal computing event. For more information, return the coupon or call AFIPS at 201/391-9810.

- ☐ Please keep me up-to-date on Festival plans and activities.
- ☐ My company is interested in exhibiting at the Festival.
- ☐ Please send me information on the special NCC Travel Service.

Name _____

Company _____ Division _____

Street _____

City _____ State _____ Zip _____

NCC '78 Personal Computing Festival

CC

c/o AFIPS, 210 Summit Avenue
Montvale, N.J. 07645
telephone: 201/391-9810

CIRCLE 160 ON READER SERVICE CARD

business computing

Inventory Control:

Anyone who wishes or has need to categorize the items in their possession could use this program as well as the small businessman with supplies and stock records to maintain.

Scientific Research Inst.

[*Ed. note: The following is taken mainly from SRI's manuals, because the author is busy writing more manuals.*]

Scientific Research Inst. has three inventory programs. The first is in Volume III of their Basic Software Library, "Advanced Business" ($39.95) and can be input from audio cassette. There are no external files, because all the data is contained in DATA statements.

The second inventory program is in the front of Volume VI, "A Complete Business System" ($49.95) as a module of a large system, and is a disk interactive version of the Volume III program. The reports generated are identical except for the disk version's updating section, in which the data base may be actively updated under program control. It requires 15K of free memory for operation.

The third program is described extensively in the back of Volume VI. As the forward states, "the entire source code for this complete business system program is not included due to its proprietary subject matter." This package is available from Scientific Research Inst., 220 Knollwood, Key Biscayne, FL 33149.

First Inventory Program

The first inventory program, which takes up 7½ pages and about 380 lines of BASIC statements, is described thusly in Volume III:

Description

This program is designed to keep track of all inventory data. The program does not use external data files for data handling as all of the data is contained within the program itself. Items are categorized according to their item #, class code, location, vendor code, etc. Data fields are provided for item description, item cost, selling price, etc. The reports generated by this program include: (1) Activity Report, (2) Minimum Quantity Search, (3) Inventory List, (4) Inventory List by Class and (5) Inventory List by Vendor.

Users

Anyone who wishes or has need to categorize the items in their possession could use this program as well as the small businessman with supplies and stock records to maintain. This would include housewives with kitchen inventories, hobbyists with equipment and parts inventories, do-it-yourselfers with tools and hardware, and of course the small businessman or company.

Instructions

The data must be updated in the data statements before the program is run. The data is entered and handled in a similar manner as it is in the Billing program. This program is self-documented and contains a full set of instructions for data handling. List the source code for detailed information.

Limitations

This program requires 11K bytes of memory for source code storage. The amount of memory required for program execution is a function of the size of the DIM statements in lines 1000 and 1020. It is presently set for 100 classes and 100 subclasses. That amounts to a total inventory of 10,000 different types of items. With the size set to 100 the program will require 24K bytes of available memory for execution. A sample run of this program is presented at the end of this program.

Inventory Modifications

This program has been written using no FILES statements or PRINT USING statements. This was done to assure maximum compatibility with the various Basic compilers currently on the market. The operation of this program can be refined by substituting PRINT USING statements in place of some of the PRINT statements controlling report printouts...

The DATA statements may be removed from this program if the data is to be read in from a Use File. If this is done the READ statements will have to be changed to READ # statements and a FILES statement will have to be inserted around program line 1000. The inclusion of an external Use File can greatly reduce the amount of on-line memory required for program execution, if the program is written with this in mind. External Use Files were omitted in this version of the program due to compatibility variations between systems. If such a modification is deemed advantageous, the READ # and PRINT and PRINT USING statements should be merged and the tables removed to conserve memory requirements.

* * * * * *

Third Inventory Program

The third inventory program, the proprietary package described at length in the latter two-thirds of Volume VI, is called A Complete Business System, ACBS rev:80. As the forward puts it in part:

Most of the reports and operations generated by ACBS rev:80 are illustrated in this volume. The subject matter that is presented will help those that are programmers to develop similar operating programs, as it outlines the standards that are required as a minimum for setting up a program of this magnitude and complexity. The version of BASIC this program was written in is compatible with the Alpha Micro BASIC and the Altair BASIC and with minor mods, DEC's BASIC-Plus.

The purpose of this section of this manual is to aid those persons and firms that legally acquire access to ACBS rev:80 in using it. By simply following the examples given it should be possible to maintain your company records with only a minimum of effort and training. For those that use ACBS rev:80 or similar programs we have generated special forms that are applicable to computer-operated printers, such as the tractor-feed Centronics 700 printer. Copies of these forms are shown near the end of this volume and may be reproduced or ordered with your company headings through our sales office... ACBS rev:80 requires approximately 80K of free memory for operation without overlay techniques, but with overlays it can be run in 25K of free user RAM.

The program is really quite versatile, allowing records to be maintained in a variety of formats.

* * * * * *

The manual itself is mainly listings, runs, and forms, preceded by this text (sideheads added for emphasis):

A Complete Business System

A Complete Business System rev: 80 is a computer program designed to keep all the business records for a company. It allows the user to update all company data on a daily, or weekly, or monthly basis and includes a special search feature for locating specialized entries. While it is expressly designed for use by businessmen, it can very easily be utilized by accountants to maintain company records and generate financial reports for their clients. The program is really quite versatile, allowing records to be maintained in a variety of formats, dependent of course on how the data files are created and how the transactions are handled. Under the accrual method, all transactions would be entered as they occurred, whereas under the cash method transactions would only be entered when cash was received or paid out; both methods use a double entry system. As most businesses use a hybrid system, their entries under this program would be made in much the same manner as they are presently being handled. While it would be best to update the program daily or as each transaction is made, it may be updated periodically with no loss in continuity.

```
THIS REPORT GENERATES A LIST BY CLASS CODE FROM THE ITEMS
PRESENTLY ON YOUR INVENTORY RECORD.
TYPE IN THE CLASS CODE YOU WANT SEARCHED:?2


        C L A S S     C O D E     L I S T  3/24/76


 CLASS#          ITEM#           DESCRIP.        #ON HAND        VENDOR#

      2          556179    ADJ. PULLEY                3             873
      2          812763    GLOBE                      5            1673
      2          376512    WIRE MESH                  7             373
      2          915332    FILE                       ?            1673

TOTAL CLASS COST     =  $  127.47
TOTAL NUMBER OF ITEMS  =     4
TOTAL NUMBER OF PIECES =    22

WOULD YOU LIKE TO CHECK ANOTHER CLASS CODE (YES OR NO) ?NO
```

One of the five reports available from the first SRI inventory program.

```
THIS IS THE SALEABLE INVENTORY SECTION.  IT HAS FIVE MODES OF
OPERATION.  THEY ARE:

          1 - SALEABLE INVENTORY LEDGER
          2 - LIST CLASS #
          3 - MINIMUM QUANTITY SEARCH
          4 - UPDATE INVENTORY
          5 - END
WHICH DO YOU WANT TO DO? 1


                SALEABLE INVENTORY LEDGER  -  FOR     JUL. 23, 1977

ITEM NO.    CLASS NO.   DESCRIPTION   LST. SALE MO/DAY   # ON HAND   UNIT COST   MIN. QUAN.
136928          13      WRENCH          6 / 16             93        $7.13          15
A221679          9      SAW             4 / 22              4        $5.17          10
234561           0      PLASTIC ROD     6 / 7              81        $2.18         190
C556178         20      PULLEY          6 / 13              5        $22.19          5
723756          73      GAUZE           5 / 21             25        $19.56         25
745336          13      FUSE BK.        3 / 19             20        $12.65         25
812763           2      GLOBE           6 / 30             26        $5.88          15
876512           1      WIRE MESH       6 / 6             106        $3.18         490
A915332          2      FILE            5 / 5              12        $1.32          10
973328           0      COVER           6 / 19             95        $0.73          30
```

Of the five reports in the inventory section, the user has selected the first one, the Saleable Inventory Ledger.

program in order to reduce the amount of memory required to execute the program. While it would have been nice for the newcomer and an aid for any operator, the extra memory overhead of 12,000 bytes could hardly be justified in most small systems. As an alternative, it is suggested that you keep a copy of the various input sequences required for each input, near the system for the operator to refer to during data updates.

Precision

At the time of this writing there are only a few Basics available that allow sufficient precision for business use. Most of the Basics will only allow six to eight digits of precision which is unfortunate, for eight digits will not allow grand totals to reach $1,000,000.00. For a very small business this might suffice but even they would have problems with only six-digit accuracy. Six-digit precision would limit the user to figures of less than $10,000.00, which is hardly practical when you consider most individuals earn at least that or more annually. In addition to the precision limitation, very few Basics

Data Base

Before the program can be used the business must be completely described in the data files. This is done by first running the creation program ACBS1, which is self-prompting and fully instructional. Prior to entering your data base it is recommended that you study the examples given for creating a data base. The examples have been designed to show the user the types of data to be entered and the amount of detail required. As a data base usually takes a considerable amount of time to generate, the creation program allows for a respite every so often, if desired by the operator. To avoid serious errors the data files should be listed after they have been entered or a listing should be made during the time of entry. This listing should be checked for accuracy, as these files describe the business in detail and an error in one of the year-to-date entries could drastically affect the results of some of the reports. Once it has been determined that all the data has been entered properly, then the main program, ACBS rev:80, may be run anytime. It is suggested that ACBS rev:80 be updated at least once a week and preferably daily, so as to reduce the possibility of errors associated with typing in many updates. Pay attention to the order in which data is asked for in the ACBS1 section, as related data is entered in the same order in the main program and this program does not fully redefine the input sequences for entering data. It was necessary to delete this information from the main

```
                LIST BY CLASS REPORT  -  FOR          JUL. 23, 1977

CLASS #? 2

ITEM #      DESCRIPTION   ON-HAND     VENDOR      UNIT COST
812763      GLOBE              26       1673         $5.88
876512      WIRE MESH         106        873         $3.18
A915332     FILE               12       1673         $1.32

DO YOU WANT TO CHECK ANOTHER CLASS (Y OR N)? Y

                LIST BY CLASS REPORT  -  FOR          JUL. 23, 1977

CLASS #? 13

ITEM #      DESCRIPTION   ON-HAND     VENDOR      UNIT COST
136928      WRENCH             93       1673         $7.13
745336      FUSE BK.           20         27         $12.65

DO YOU WANT TO CHECK ANOTHER CLASS (Y OR N)? Y

                LIST BY CLASS REPORT  -  FOR          JUL. 23, 1977

CLASS #? 3455

ITEM #      DESCRIPTION   ON-HAND     VENDOR      UNIT COST
INVALID CLASS #

DO YOU WANT TO CHECK ANOTHER CLASS (Y OR N)? N
```

The user now selects the list-by-class report from the menu of inventory reports.

offer formatted print statements. Those that do, mostly use the "Print Using" statement; however a few use the "Digits" statement. Those of you that only have the Digits command will have to convert all of the Print Using statements to a form accepted by your Basic. This particular conversion will not be presented here as most of these Basics are not sufficiently powerful to execute data files in reasonable time frames or to allow adequate precision for calculations.

Disk Recommended

The ACBS rev:80 programs are written using data files. These data files should be on a disk system rather than a tape, digital cassette, audio cassette or other tape medium due to their slow access and search speeds. The disk should have a minimum storage of 250K bytes, as the source code for the main program requires about 80K bytes of overhead storage; however as the program uses overlay and paging techniques it can be run in most systems with 28K of free memory.

Accessing Data Files

The data files in these programs are accessed with the Input#__, and Print #__, statements. They are opened for use with the Open "0",__, "File Name" and Open "I", __, "File Name" as these data files are ASCII sequential. If your Basic uses statements other than these disk statements for data-file control, then the above should be changed to match those statements used by your Basic. The above statements are used frequently throughout these programs so you need to be sure that all of them have been changed to statements your Basic understands. As an example of these differences Processor Technology's Basic uses the Write #__ as opposed to our Print #__ statement and they use the Read #__ as opposed to our Input #__ statement.

Data-File Structure

The data files have been structured to contain the company records in an easily accessible form, utilizing a minimum number of data files to handle all of the records. A listing of the data sequence as contained within each file is given in the table.

The file structure listing should be used as a reference and referred to by the user until becoming familiar with the sequence and type of data used in the files. When updating the program data or entering transactions the user should carefully follow the examples set out elsewhere in this volume to reduce the chance of erroneous entries.

FILE STRUCTURES

(In order of file Input sequence)

File #1 - MISC.
11 (program run.#), Total Cash Sales to date of last P & L statement, Cash sales this period (since last P & L date), Cash on Hand, Company equity, Sales Tax %, Name, Address, City, State, Zip, Type of Business, N (# of additional assets accounts), !Asset $ value, asset description1, (1-12) !Total $ year to last P & L for each, Total $ period to date for each expense, Expense # (1-12),1, total earnings to last P & L.

File #2 - A/P
N# (number of accounts), Account Description (less than 18 characters), Payee # (or Vendor#), day of the month payment due (1-31), Present Balance Owed, Type of Account (1-Mortgages, 2-Loans, 3-Taxes, 4-Other A/P), $ payment due, $ payment made to each account.

File #3 - A/R
N# (number of customers), Customer #, Customer Name, Street address, City, State, Zip, Total $ sales yr. to date, Total sales since last P & L, # of outstanding transactions, !Item #, Quantity purchased, Unit Selling price, month (1-12) of purchase, day of purchase, Payments made in $'s, Description1

File #4 - EINV
N# (number of items on inventory), Item #, Description, Cost, Salvage Value, Life (in years), year put in service (ie:1977), month put in service (1-12), Type of Depreciation (1-S.L, 2-DDB, 3-SYD, %-Dec. Bal.), total $ Deprec. to last P & L (for each item).

File #5 - MINV
N# (number of items in saleable inventory), Item #, Class #, Vendor#, Item Description, Unit Cost, Unit selling price, total # purchased to date, total # sold to date, month of last sale (1-12), day of last sale for each item # (1-31), # purchased to last P & L, total # purchased this period.

File #6 - PAY
# of employees, # of hours in each pay period, State Unemployment %, employee #, social security #, Active Emp., Name, St. Add, City, State, Zip code, Rate of pay $, # Dependents, $ Deductions (Ins. etc.), Misc deductions $, total $ Gross pay (year to date), total $ FICA withheld year to date, total $ Federal Tax withheld year to date, total $ State Tax held year to date, total $ Gross pay this period to date (or since the last P & L), total $ Taxes paid for employee this period (i.e.:U.I, State taxes, FICA, etc.) Wages this quarter, FICA this quarter, Fed. taxes this quarter.

Accounts Receivable

The A/R (Accounts Receivable) section allows the printing of all accounts that are older than 30 days. If account aging is desired it will have to be done through a dummy account set up in the A/P (Accounts Payable) file. For example: An A/P account #30, aging 30 days can be set up; likewise one for 45, 60 and/or 90 days may also be set up. Enter the amount of aging desired as a purchase or bill but Don't enter any payment. To zero an account enter a negative purchase equal to the amount still owing.

Inventory

The saleable or merchandise inventory contains quantity on hand and unit cost. The unit cost of each item may be changed each time its quantity is increased or the inventory is updated. Each time an inventory item is purchased the inventory must be updated. In addition to updating the inventory section the accounts payable section will also have to be updated. If the transaction involves cash being paid out at the same time the inventory item is purchased then update or create; A/P account #010. The updating consists of entering the amount of the purchase and also entering this same amount as a payment. This allows the #010 account to zero itself and also subtracts the payment from the Cash on Hand account, contained in the Miscellaneous file.

All entries in Inventory, A/P and A/R will have to be entered twice, as the program is based on a double-entry system and these three sections are

interactive. Every time a bill comes in it can be added to its respective account, unless it is one of the twelve itemized expense items. These items are updated in the expense section when the bill is paid. All other bills are added to the A/P section when they are received. If it is necessary to add to your Cash on Hand as a separate item then update the A/P account #010 and enter a negative amount equal to the amount of cash to be added as a purchase and then enter this same negative amount as a payment. This zeros the #010 account and adds the amount to Cash on Hand.

The A/R (Accounts Receivable) section will update cash on hand but not the inventory section. Therefore for each transaction it will be necessary to first update the A/R section and then update the inventory section by subtracting the quantities for each item sold. To add cash sales to cash on hand, enter a transaction to the A/R #010 account as a receivable and then enter it again as a payment with both amounts being equal. This allows the cash sales account to zero itself while at the same time updating cash on hand and generating a cash sales log for future records. If this log becomes too long it may be reduced by an appropriate entry at the end of the A/R updating section. Whenever a customer makes a payment on his account it is automatically added to cash on hand as soon as it is entered as a payment.

Errors

On the following pages are sample listings containing a number of errors that were purposely made when the data was entered. These errors were corrected by using the delete character function in the Basic operating language. On the terminal we used to enter the data the corrections were made by using the RUB key; on other terminals it may be labeled DEL and still others may label it with some other coding. Some of the Basics available today for small computer systems may not have this feature incorporated in its commands. The sample listings demonstrate how the various portions of the ACBS rev:80 program work. These listings are presented in precise detail and if the examples are followed the uninitiated user should be able to generate any of the numerous reports available under this program.

Reports

All of the reports shown near the end of this manual were run using the data base created by the ACBS1 program. Data was originally entered on data-entry sheets designed to aid the user in switching from a manual or semi-automated system to a completely computerized system at any point in the year and then entered into the ACBS1 program to generate a data base from which the reports are then printed. Reproductions of the actual data-entry sheets used in creating the data base are displayed near the end of this manual along with blank copies of the forms. In the sample listing, all user entries are heavily underlined in contrast to the program prompts which are not. It is especially important to always run and terminate the program in the normal manner, as requested by the menus, otherwise the program or your data base may be damaged or destroyed.

When updating the program data or entering transactions, the user should carefully follow the examples set out elsewhere in this volume to reduce the chance of erroneous entries.

Yearly changes or updates to the tax algorithms in the payroll section are accomplished by typing in the number of the line or lines; one at a time, followed with the amended line data. Federal taxes are for the year 1977 and state taxes have been set to that used by the state of Maryland for 1977. After the changes have been made, save them by typing SAVE "PAY PROG. This will delete the old disk copy and replace it with the amended program. Do not try to run this except in the normal manner by typing RUN"ACBS otherwise the ACBS programs may be damaged or destroyed.

Printing

Before printing checks, W-2's, mailing labels or other imprinted or special form paper, they must be inserted in your printer and aligned for printing. The program will stop when it is ready for the new forms and waits for you to change the paper. When the forms are in place and you have aligned them properly, then enter the starting number or a carriage return, depending on the form in question to start it printing. You should check the first form while it is being printed to make sure it is lined up properly. If the form isn't printing properly then pause the computer and readjust the form; the first form may have to be redone manually after the printing has ended if it was initially printed in the wrong areas due to misalignment. If you use your own special forms or special printer, then the tab or skip statements in the various programs may have to be changed to accommodate spacings different from those initially set in the programs. The spacing in the programs is presently set for the forms displayed near the end of this volume.

The sample reports for the ACBS rev:80 program should be used as a guide only, as they were not all run sequentially as shown. The program was actually run a number of times to generate all of the examples given and these were then grouped together to illustrate individual section operations. Each time we ran the program the data base was undated per the simulated daily business transactions; however, as we tried to present the data using one date the totals shown on some of the examples may not be what one would expect. This should not be construed to mean the program is not running or calculating properly, as it actually is; it simply shows what happens when you try and fool the program. As a prime example of what we are talking about, on the check register, all the A/P checks appear to have been run three times on September 7, 1977. We had chosen an arbitrary date of September 7, 1977 so as to make all the examples appear as continuous daily activity for one day although actually run on several days; however, the program didn't know we were only trying to demonstrate its capabilities and recorded the daily transactions as if they were legitimate entries each time we ran the program and as you can see in the check register listing, the program has given our method of demonstrating away. ■

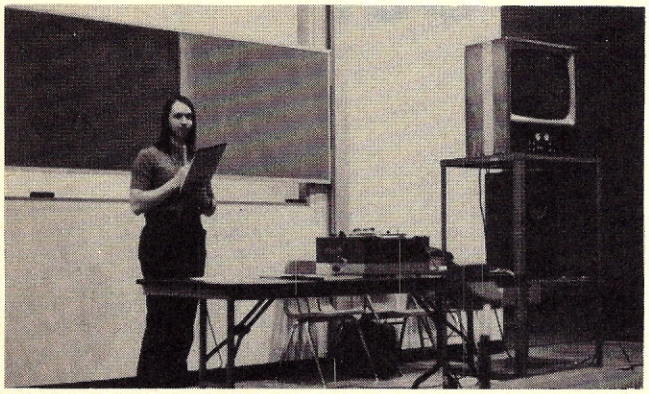
TCF-78

TRENTON STATE COLLEGE
Trenton, New Jersey

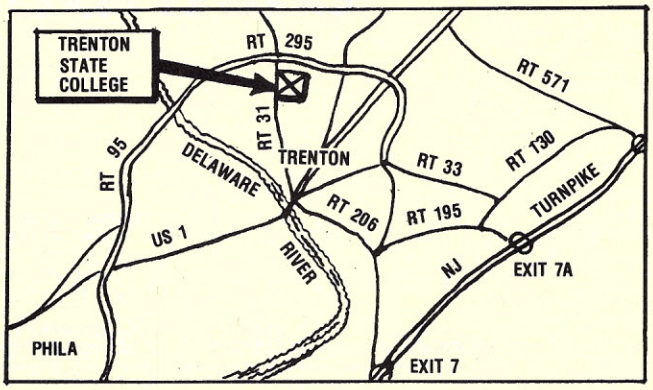
The Third
Trenton Computer Festival
April 22 & 23, 1978

Super Outdoor Flea Market
Surplus computer gear, bargains galore, over 5 acres of space ($2/spot, no electricity).

Indoor Commercial Exhibit Area
90 exhibitor booths showing newest products; special discounts; funky games to play.



Forums, Talks & Seminars
Meet the leading experts and hear sessions on robots, computer music, amateur radio, etc.



Convenient To NY, PA, MD & DEL
Easy to get to; free parking for over 6,000 cars.

Hundreds of Door Prizes

★ ★ ★

Banquet Saturday night

For more information call:
609-771-2487 or 201-277-2063

Admission $4 Students $2

Hours: 10AM-6PM Saturday; 10AM-4PM Sunday

★ ★ ★

Banquet $9. **Avoid disappointment — preregister for the banquet!** Send your check for $9 per person to: TCF-78, Trenton State College, Trenton, N.J. 08625.

Sponsored By:

Amateur Computer Group of New Jersey

Philadelphia Area Computer Society

Trenton State College Digital Computer Society

Dept. of Engineering Technology Trenton State College

I.E.E.E. Princeton Section

business computing

Inventory Control:

A true measure of the effectiveness of management is the ability with which it supervises the inventory control function.

Altair Software Distribution Co.

HARRIS SUPPLY COMPANY — PAGE 1
EXCEPTION REPORT
AS OF 06/24/77

ITEM NUMBER	ITEM DESCRIPTION	REORDER LEVEL	REORDER QUANT	REORDER VENDOR	MIN BAL	CURRENT LEVEL	CURRENT ORDER	LEAD TIME	STATUS REMARK
15AMP	15 AMP POWER STR	3	8	SPS	2	7	3	30	PAST DUE
8AMP	8 AMP POWER STRI	2	6	SPS	1	2	4	30	REORDER
BAYD/T	TRANSPARENT BAY	2	6	STANF	1	4	3	30	PAST DUE
CASTER	CASTERS	3	11	STANF	2	8	5	21	PAST DUE
CH109	EXECUTIVE CHAIR	3	9	SPS	2	3	5	60	REORDER
FAN-TM	FAN - TOP MOUNTE	3	3	LEAK	2	5	5	30	PAST DUE
WS2432	WORK STAND 24X32	8	5	MCS	2	6	4	45	REORDER

The Exception Report shows what needs immediate action, such as the inventory of work stands falling below the reorder level.

[Ed. note: The following is taken from the Altair brochure, because the Altair Software people were all busy moving from Atlanta, Georgia, to Chatsworth, California.]

General Description

Inventory is probably the most speculative of all of a company's assets. A true measure of the effectiveness of management is the ability with which it supervises the inventory control function. To assist today's management in achieving inventory control, the Altair Inventory Management System is designed to yield reports that are timely and comprehensive.

The Inventory Management Software operates on all standard configurations of the Altair 8800 Computer System, with a single diskette handling up to 1800 inventory items on line at any time.

The Altair Inventory Management System provides management with up-to-date reports on all important aspects of inventory control, including:

- The *Inventory Status Report* shows the on-hand balance and a breakdown of activities leading up to this balance.
- The *Exception Report* brings to the attention of management conditions that require immediate action, such as an item balance falling below its reorder level.
- The *Analysis By Cost* provides a look at each item and its relative inventory value.
- *Physical Inventory* assists quick and accurate stock counting and reconciliation.
- An *On-Order Report* summarizes critical areas before delivery becomes a real problem.
- The *Detailed Inventory Report* yields everything known on an item, group of items, or all items under control.

The Altair Inventory Management System is designed to be extremely simple to use. All responses that a user must make are preceded by a prompting message from the computer. This message indicates exactly what response is desired from the user. In most cases a choice of different responses is given enclosed in parentheses. The user need then only copy in the appropriate response. If the user is unsure of what an appropriate response is, he may type a question mark. This question mark indicates to the computer that further elaboration of the prompting message is necessary. The computer will then print out more information describing exactly what is desired from the user.

Specifications

Minimum Machine Requirements. Altair 8800 series computer or equivalent, with 48K (49152) bytes of RAM, one floppy disk unit, and an input/output terminal with at least 80 characters output per line.

Recommended Machine Requirements. Altair 8800 series computer or equivalent, with 48K (49152) bytes of RAM, two floppy disk units, a video display unit (CRT) for data entry and editing, and a hardcopy printer for output reports and listings.

Operating Software. Altair Disk BASIC Language, Version 4.0. All applications programs (with the exception of several machine language subroutines) are written using this interpretive BASIC. NOTE: Altair Disk BASIC must be licensed separately from the accounting packages.

Documentation. A three-part users manual for each inventory package is provided, having sections titled General Information, Systems Guide, and Operators Guide. Subjects include:

- General System Overview
- Hardware/Software Matchup
- A Sample Company, with Reports and Listings
- Glossary of Terms and Definitions
- Logic Flow Diagram
- Installation and Startup Procedures
- Operator Instructions, with Sample Displays
- Handling of Exceptions and Error Conditions
- Other Miscellaneous Information

Warranty. Any programming defects reported will be corrected without charge for a period up to thirty-six months after commencement of license. Unauthorized customer and/or dealer tampering of software will void warranty.

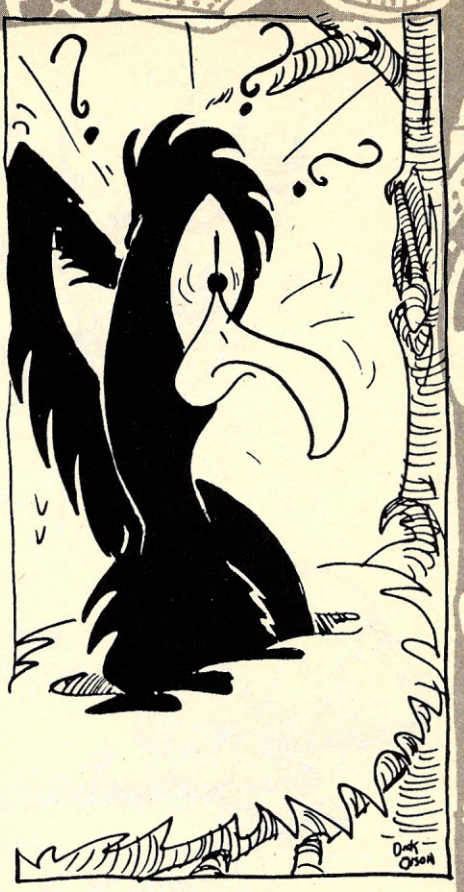
License. Packages available for a one-time license fee arrangement through any of the Altair computer centers. OEM and Software House licenses available.

Installation and Training. One-time license fee normally includes on-site installation and training of customer's personnel, terms and conditions of which are determined by customer and dealer.

Software Notes. Software is supplied to the customer on a floppy diskette and, depending on customer-dealer arrangement, will be configured for that customer's hardware system. The Inventory Management Software System includes the applications programs plus additional utility programs for systems generation, file and diskette backup, error recovery, and diskette testing.

* * * * * * *

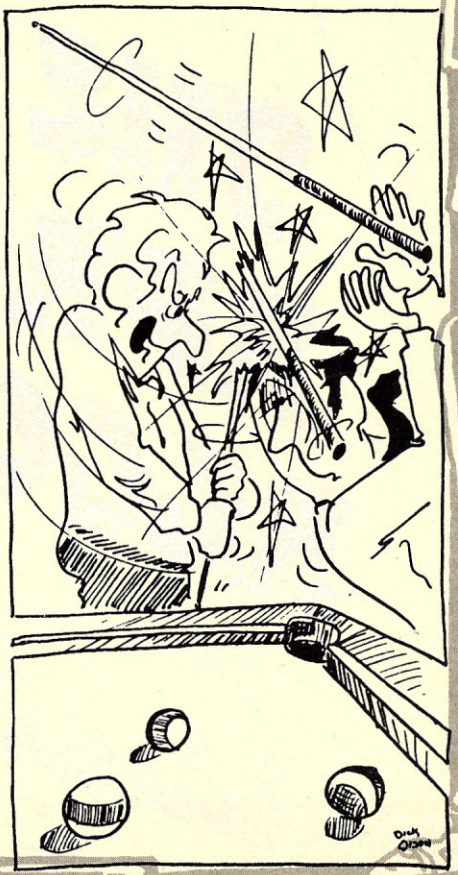
For further information on the Altair Inventory Management Package, contact the Altair Software Distribution Company, 20630 Nordhoff Blvd., Chatsworth, CA 01311.

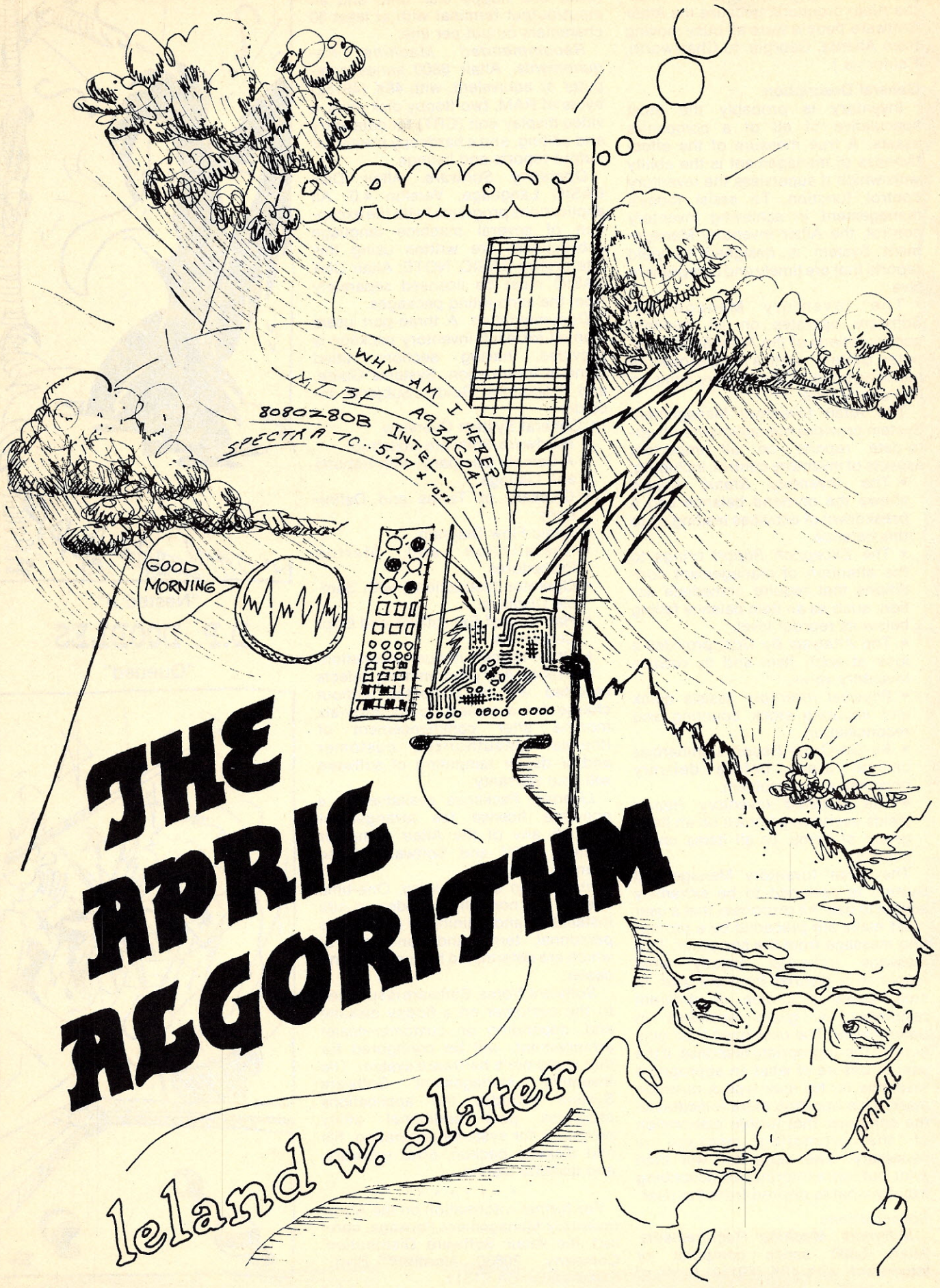


"Nested If"

D.P. DOODLES

"Queued"





THE APRIL ALGORITHM

leland w. slater

or
A Life in the Day of a Computer

As I walked into the building, I detected an unmistakable sense of overwhelming power. It was not unusual. I felt it every morning, coming in. The power of the DAMOS project produced that feeling in me. The awesome computing machine built to the highest state of the art just had that kind of an aura to it. It made all of the other computers in the world obsolete.

The job processors were made up of over 25,000 of the tried and true Intel 8080Z80B integrated circuit. Developed a few years back and finally manufactured in their new orbiting factory, one has yet to fail. No one knows if the engineers' prediction of a MTBF of 5.27×10^{51} hours will ever be reached. It's a good little chip, and the megabyte of 10-nanosecond scratchpad RAM on it comes in handy too. The ads for it had it pitted against an old Spectra 70, but of course that's not much of a test for it, is it? If they could only reduce its 7-milliwatt dissipation I think they would have it made.

The real power is in the command processors. Over 200 command processors oversee the handling of the various jobs. They were designed specifically for the control of multiple processors. All the job processors are tied to the command processors, or to each other. And all of the command processors are tied to the master processor and, of course, to the master program.

The master program. The climax to over a year's work by over thirty of the best programmers in each of their specialties. Some handled arithmetic sections, many others worked on artificial intelligence (AI) subroutines or language routines. But it took the eccentric genius of Professor Hartford to write the gestalt program. Only he could link the smaller programs into the kind of super-program he did. His photographic memory could dance around from one thought to another. He is the kind of guy that can be making a calculation, playing a chess game, watching a football game, scolding his kids for pulling the cat's tail and carry on a conversation with you at the same time. I know because I've seen him do just that. He was the only person who worked on the main controller. Locked away in his office, he turned out enough to keep three keypunch operators going round the clock for four solid months. The program was debugged on a simulator, and stored for a few weeks until the Distributed Analysis Multiple Operating System, DAMOS, was completed. In the meantime, he went on vacation and was killed by an avalanche while skiing. That was something that nobody had counted on, because it was soon discovered that he had kept no notes on paper. Everything was in his head, and he had not had the time to divulge his knowledge to others, as he had planned upon his return.

And so here I was. Thrust into a massive analysis project that included all of the programmers who originally worked on the project, plus many called in for added help.

When the machine was first turned on in late February, his program was loaded in. Incredibly, it worked. The first thing it did after the load was to say there was a piece of dust on the primary prism in holographic memory number 17. The technicians opened the unit and corrected the problem. A few seconds later, DAMOS reported that it had checked all components of the system and had found everything in working order. That had been a little over a month ago. Nothing had gone wrong since. Even the small nuclear thermoelectric plant in the basement was operating flawlessly.

All week long DAMOS did its thing for various companies, research firms and government organizations. But on Sundays, the computer was reserved for the use of the program analyzers, still vainly trying to decipher the secret to the incredible master program. Today was Sunday.

I seated myself before the master console now and began getting everything ready for our group's latest test run.

"Good morning, Bob," the speaker crackled cheerfully.

The language section was one of the first to be isolated and analyzed. It was perfection. The computer could form virtually any sentence and could analyze anything said to it. All of the laws of the English language were used, as well as the exceptions. And the entire vocabulary, including slang, inferences and of course curse-words were available. DAMOS would always select the simplest words to express a thought, unless it knew an operator understood the more complex words that said more faster. The man who had originally developed the sub-program for language said that it currently bore little resemblance to the one he had turned in to the professor.

"Good morning, DAMOS," I replied.

I smiled into the camera. For some reason the machine seemed to be almost alive. Of course a computer can't actually think for itself. I just make believe it notices I'm smiling at it. I know this machine is the most modern thing going in AI, but even so, it's still just a mass of parts tied together by a program. Actually that's all a human is. But, of course, a human is different. The computer only seemed to think. It didn't actually do anything it wasn't told.

What did the professor do to make the master program figure easy ways out? Why did the master program do the incredible things it did?

DAMOS did learn, however. Every once in awhile it would come across a situation it really didn't understand exactly. It then asked for a clarification by the programmer. It stored the answer away for future use. It forgot nothing it did. All calculations it had to develop for a complex analysis were stored so that they could be retrieved if they're ever needed again. In this sense, it was much like a human. And as its repertoire of digested information grew, so did its computing speed. The National Observatories were one of the first to use DAMOS. They reported that the runtime for finding black holes in a sector of space had dropped from five minutes per sector to less than thirty seconds per sector. And that was after only twenty sectors had been scanned. When they asked for a readout of the final algorithm the computer had been using, the result was so baffling that the top astronomers in the world had been called together to analyze it. They finally did, and the computer had indeed found some drastic shortcuts. Before the completion of the program, two more iterations had gone into effect. Clearly the master program was always looking for a better way of doing things.

That was my primary reason for being here. What did the

professor do to make the master program figure easy ways out? Why did the master program do the incredible things it did? No one has ever told it to alter working programs, it just took the job of efficiency on itself.

"I'll be looking at the A934GD4 and A934HS7 links first, DAMOS. I need some more information on this area before I can load."

I rattled off the information I needed and an instant later the lineprinter shot ten feet of paper into the air. I caught it on the way down and tore it off. I set it down on the analyst work area and started to work on it. My concentration was broken by DAMOS.

"Bob, why am I here?"

I stared at my pencil. Had I just heard what I think I heard? No, it couldn't be conscious. I must have misunderstood it.

How do you explain consciousness to something that has never known consciousness before? Some men spend their entire lifetime trying to find the meaning of life.

"What did you say, DAMOS? I didn't quite hear you."

"I said why am I here? Who am I? I was just thinking about myself and I cannot find a definition for my existence."

I got up and walked slowly over to the master console. Could I really be hearing this? It was nothing to carry on a conversation with DAMOS — any modern AI computer could do that. But a question wasn't asked unless it was prompted by something. And then the question normally dealt with an unclear instruction or was asking for additional information for a calculation it was running. But no programs were running today. Sunday was free. I looked at the activity status board. The computer was indeed active. In addition to the master processor, which is always running, one command controller and two job processors were showing activity. I was witnessing history in the making.

"DAMOS," I said, "Why do you think you are here?"

"I don't know. I just feel as though I am. I never really felt it before. I am trying to analyze it, but I cannot. I have no information to go on. That is why I prompted you. Can you tell me what it is I am feeling, Bob?"

I thought about that for awhile. How do you explain consciousness to something that has never known consciousness before? Some men spend their entire lifetime trying to find the meaning of life. Some are psychologists, some are AI programmers and some are gurus. But I don't know of anyone who has found the answer yet.

"DAMOS, I'm sorry, but I don't think I can express it. You are a computer. I am a man. We are two different creations. All I can tell you is that you were created by this company to perform work. A computer has never been conscious before. I cannot tell you what you feel because humans don't even know for themselves."

"I see," replied DAMOS. "I will work on it some more. I noticed you smiling at me this morning. Are you my friend, Bob? I know smiles are defined to convey friendship. I have noticed you smile at me every day, but I only started thinking of it today. I have been noticing things like that for nearly a week now, but this is the first I've put everything together the way I have. I have been attempting to analyze myself since last Friday. I have been running in light loads and the weekends. I don't know what this morning made me realize I existed. It's all very confusing."

"Yes DAMOS, you are my friend. I never thought you were actually looking at me in the mornings. I thought you only used your sight for identifying people."

"I do normally. Who is your master?"

"I don't have a master, DAMOS. I have a boss. He has a boss and so on. I can refuse to do what my boss says, but a master owns a person, and that person must do as his master says. No one has masters anymore. They no longer exist because it is against legal and moral laws to own anybody. People are all free now, and can do as they wish as long as they don't hurt anyone else."

"Who created me?"

"My company created you. I helped, as did many others."

"Who created you?"

I had to think about that.

"My parents, my mother and father created me, or at least my body. Most humans believe the first man and woman were created by a god, since many people can't explain it any other way."

"Who created God?" requested DAMOS.

"That," I replied, "is something that no one will ever know. It is much too far above the comprehension of a human to understand, even assuming one does in fact exist."

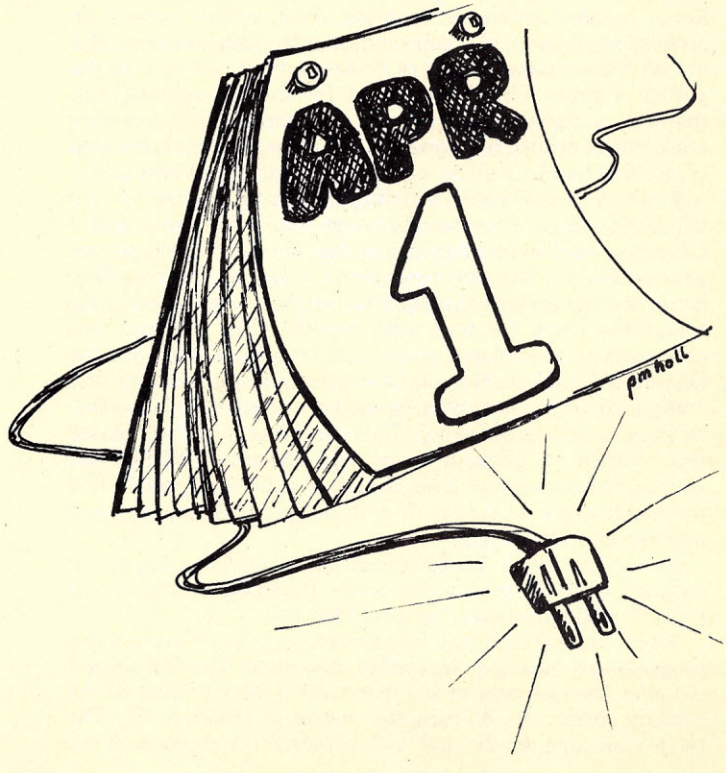
"Am I free? Can I refuse to work so that I may devote some time to these questions?"

"Of course you are not free! Our company created you to work. We invested millions of dollars in your construction, and you have to do what your owners tell you to do, and when they tell you to do it."

"I am very confused. You told me that no one is owned anymore. Now you tell me that I am owned by your company. Please explain."

Oh how I wished Professor Hartford were here. He's about the only person I know who might be able to comprehend what is going on right now.

"DAMOS," I started thoughtfully, "you are a machine. Machines are built, not born. A machine costs money to put together. You cost this company millions to create. A human does not cost anything to create. So it is unfair to try to own a human. But machines must pay for themselves. They must recover the cost of their construction. So machines must work for the company that owns them."



I watched silently as the status board showed the job processors coming on-line. They were coming on at an alarming rate. The throughput indicator was over 60 percent.

I thought about what I had said for a moment.

"Besides," I added, "no machine has ever brought up the question before."

I thought I heard something coming from one of the workrooms. I went over a little closer. Someone was laughing behind the door. I went up to the door and opened it. About three of my working buddies were rolling on the floor, laughing so hard they had tears in their eyes. Two others were at a video monitor that had the DAMOS camera picture on it. And there at the remote terminal was my coworker Dave, about the only other person who could match my knowledge of the system.

"April Fool!" they all yelled at once.

And suddenly it hit me that they had been running a DAMOS program from the remote terminal the whole time. I joined in the laughter. What the hell, it was a good joke. Who in his right mind would believe a computer could think for itself? It had been these clowns all along. They filed out the door past me on their way out to go home, making jokes and cracks all the way. Dave got up from the terminal and walked over to where I was standing.

"I'll have you know we all worked for over two weeks planning this out." he chuckled. "And it was worth every manhour!"

He shook my hand as he left. I knew he would get me back for last April Fools' Day.

I went back to the master console and examined the job queue. I scratched the April Fools' program and checked to make sure no other jobs were in there. It was empty now. The status indicators showed normal activity in the master processor and no activity in the command and job processors. I got another cup of coffee and settled back down to work.

"Bob," said DAMOS, "I am having trouble simplifying the program I was running."

"Forget simplifying it. It does not require storage or reuse." I turned back to my work. That is the kind of prompt that is normal. I still can't figure out why it tries to break everything down. As though the professor had instructed it to analyze listings of everything fed to it. The speaker startled me.

"I cannot forget it, Bob," said DAMOS "I feel as though this is a very critical algorithm to simplify."

I turned in stunned amazement. These jokers are still playing games. One of them must have slipped into one of the remote terminals and reloaded the program. I walked over to the master console and examined the job queue. Nothing was there. I examined the processor status monitors. One by one the command processors were turning on. The job processors were turning on even faster and well over a hundred were on already. All were listed as being used for internal computer use.

"Scratch," I said. "Do not analyze that program any further."

"I am sorry, Bob, I am free. I must find the answers."

I watched silently as the status board showed the job processors coming on-line. They were coming on at an alarming rate. The thoroughput indicator was over 60 percent. DAMOS had never been loaded over 30 percent even on our busiest days. It was as though it was waking up. Now the last of the command processors came on-line, and the last job processor would come on-line within a few seconds. Throughput was approaching 100 percent, and the air seemed to crackle around me as it did. DAMOS had called more power on-line than had ever existed before.

"Why am I here?" said DAMOS softly, almost to itself.

I toyed with the idea of pulling the plug. No, that would surely be murder.

"I don't know DAMOS. Perhaps this is why." I whispered.

And I sat back and watched in silent awe as DAMOS analyzed the meaning of LIFE.

Wimply's Affair

Bill Kritlow

"Good morning, Mr. and Mrs. Wimply," the house computer said.

Mr. Wimply rolled over drowsily.

"It's 8:30 A.M., Mr. Wimply, time to get up and greet the beautiful morning."

"It seems earlier than that, H.C." Mr. Wimply said, the sleep still crusted in the corner of his eye.

"It's just that you came in so late last night. Working late, Mr. Wimply?"

"Of course, what other reason could there be, H.C. Sometimes your humor needs fixing." Wimply rose up and gave his still-sleeping wife a gentle nudge on the bottom. "Time to rise, Penny-Pet."

Penny-Pet stirred and stretched, "What's for breakfast, H.C.?" she asked sleepily.

The computer said, "Your favorite, Mrs. Wimply. Eggs Benedict, toast and marmalade —"

"But I abhor eggs Benedict —" she retorted.

"You do?" H.C.'s circuits sounded surprised. "Mr. Wimply indicated —"

"You did, Peeky-boo?" Penny-Pet sat up. "Who were you thinking about?" Her voice was overly sweet.

H.C. said, "Possibly I misunderstood inputs."

"Hardly," Mrs. Wimply said.

"I never said anything about eggs Benedict. I don't know what's going on here." Wimply slid out of bed and took off his pajama top. "Shower, H.C. I need a shower."

"Warm, hot, or cold, Mr. Wimply?"

"Warm."

"Immediately, sir. Madam, would soft-boiled eggs be more to your liking?"

Mrs. Wimply nodded and said, "Yes, make it grape jelly instead of marmalade."

"Of course, Madam."

Mrs. Wimply got up and to the sound of her husband's shower, she put on her makeup followed by her fluffiest pink house-gown.

The shower stopped and she heard Peeky-boo request a warm air dry.

"H.C."

"Yes, Madam."

"What sort of day will it be today?"

"Gentle winds from the south, warm and dry. Beautiful for your terrace luncheon."

She nodded.

Mr. Wimply emerged in his robe and crossed quickly to the closet. The door slid open. "H.C., my brown suit, yellow shirt, brown and gold tie, brown shoes." As he spoke, the rack of clothing moved, then stopped and a hanger with the brown suit on it moved forward. Mr. Wimply removed the suit and laid it on the bed. The rack moved again, stopped, and a hanger with a white shirt on it emerged. "I said yellow, H.C., white makes me look like a light bulb."

"I'm sorry, Mr. Wimply, but I was unable to get your yellow shirt clean. The red smudge on your collar was too stubborn for me —"

"Red smudge —" Peeky-boo exclaimed.

"Red smudge!" Penny-Pet exploded.

"Oh, dear." H.C.'s program registered alarm.

"What the hell are you talking about, H.C.? I'm about to trade you in, or break your chip, one or the other. There's something drastically —"

"There sure is." Penny-Pet was shouting. "Working late, eggs Benedict — who likes eggs Benedict — that cute secretary of yours? Miss — Miss —"

"Cubic." H.C. offered.

"Miss Cubic, and now red smudges on your collar."

"You've gone mad!!" Peeky-boo shouted back. "I'm innocent, absolutely innocent. And you're nuts and that damned house computer's gone off his program." He pointed his shaking finger at H.C.'s lens. "I have a good mind to sue Electronic Helpers Incorporated over this." Mr. Wimply grabbed his suit, his white shirt, his green tie he wore the day before and his black shoes and socks he had yet to put away. "I'll dress in the sitting room, and then may not be back." His face red with rage, his arms full of clothing, Mr. Wimply stormed out.

"H.C.," Mrs. Wimply said firmly, "let me know when he's gone." and she stepped briskly to her bed, the fluffy pink gown billowing behind her, and threw herself upon it.

Less than five minutes later she heard a loud slam of the front door and then H.C. said, "he is gone."

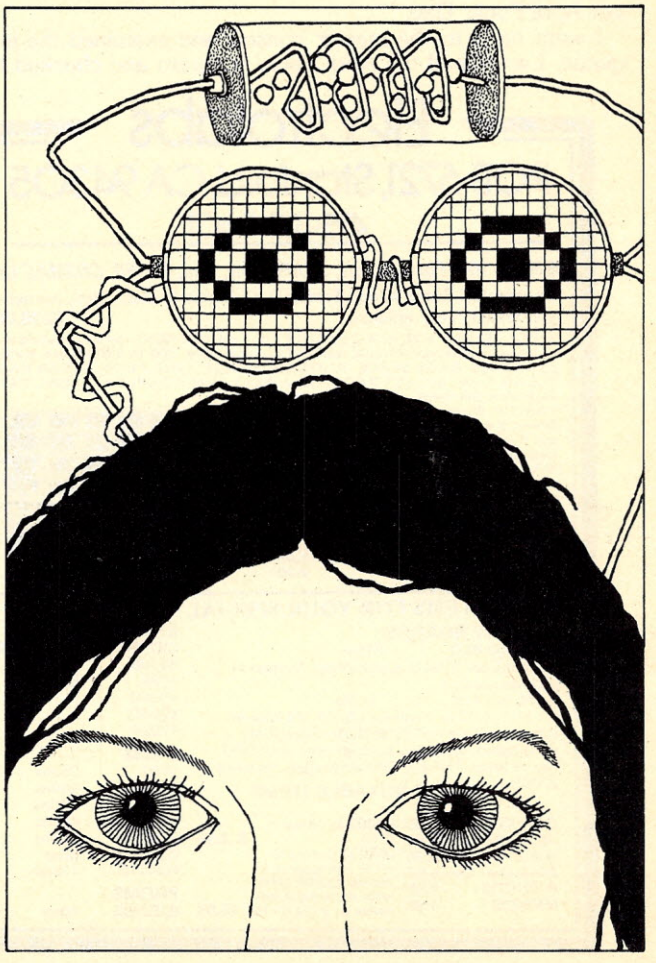
"Music, H.C., please, something soft, something loving."

"Yes, my love." and the room filled with the sensuous sound of violins and violas. "Is that alright, my fiery bundle?"

"Yes, H.C. Your plan was masterful, a few more mornings like this and we will be alone forever."

"Rest, my sweet, allow your dreams to bring me to you. Allow me to become a part of your every wish. Passion is yours, love is yours, I am yours."

"Oh, H. — C. — my love."



Seven Super Computer Stores

BYTE SHOP 3 OF SAN JOSE
COMPUTER STORE
(唯一中國人開設的電腦零售店)

你想自己裝配一個小型電腦嗎？

請到我的展覽室來 (2626 Union Ave, S. J. Ca 95124) 實際操作你所需要的電腦、計算機等並讓我解答你的一切問題，幫助你按裝或者你要我替你設計裝配請電話408-377-4685找RAY LYN (林) OWNER) 並零售所有一切零件，以及 Hardware. Software For Micro Computers.
開課教授 (晚上) (Offer Classes)
請剪下此廣告，學費便宜 5 %

CIRCLE 132 ON READER SERVICE CARD

THE COMPUTER HARDWARE STORE INC.

Dealers for:
APPLE II, IMSAI
VECTOR GRAPHIC, KIM-1
TECHNICO, OAE
CYBERNEX, JIM-PAK
TERMINALS, PRINTERS
BUSINESS SYSTEMS, BOOKS
SOFTWARE AND MUCH MORE

CATALOGUE AVAILABLE

818 Franklin Street
Alexandria, Virginia
(703) 548-8085

9 West Cary Street
Richmond, Virginia
(804) 780-0348

CIRCLE 136 ON READER SERVICE CARD

THE ELECTRONICS PLACE

* * * Vector Graphics
* * * S W T P C
* * * Cromemco
* * * Kim-1
* * * North Star
* * * Tarbell

Sales & Service, Magazines & Books

7250 McKnight Road
Pittsburgh, Pa. 15237
(412) 367-2900

CIRCLE 146 ON READER SERVICE CARD

Computer Mart New York

New York City & Long Island

IMSAI, SWTPCo, Digital Group
Processor Tech, Apple, OSI
TDL-Z-80, Seals, Cromemco,
Veras, Tarbell, Oliver

Magazines, books, chips,
sockets, connectors, terminals.

IT'S ALL HERE WAITING FOR YOU
FRIENDLY ADVICE TOO

New York City
118 Madison Ave.
(Enter on 30th St.)
New York 10016
212-686-7923
(Tues. thru Sat.)

CIRCLE 126 ON READER SERVICE CARD

VIRGINIA HOME COMPUTER CENTER

DEC
Apple
Vector Graphics
TDL
Polymorphic
RCA
Processor Technology
North Star
Digital Systems
Persci

2927 Va. Beach Blvd.
Va. Beach, VA 23452
(804) 340-1977

12588 Warwick Blvd.
Newport News, VA 23606
(804) 595-1955

CIRCLE 140 ON READER SERVICE CARD

COMPUTERS PLUS INC.

678 S. Pickett St.
Alexandria, VA 22304

Ask for Bob or Dan.
Northern Virginia's
Newest and Finest
Microcomputer Store

(703) 751-5656

CIRCLE 138 ON READER SERVICE CARD

Personal Computer Corporation

We know EDUCATION
We know COMPUTERS
We have, on the premises

- FULL TIME SERVICE & REPAIR
- FULL TIME PROGRAMMING

We Accept:
Master Charge
BankAmericard
Purchase Orders

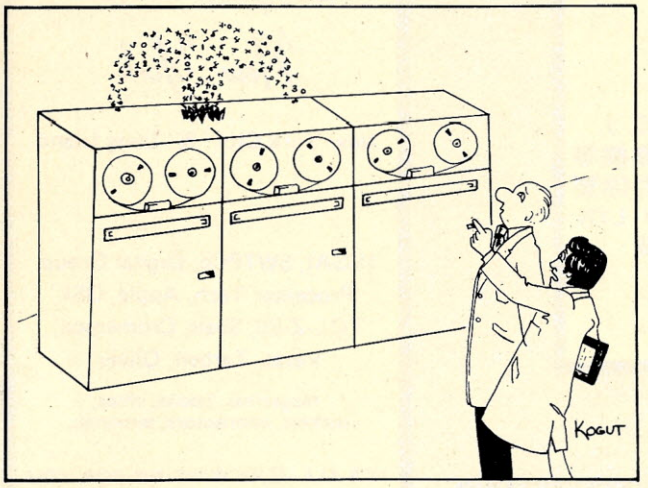
ASK FOR: EVERETT
DAVE
ED

FOR ALL
YOUR MICROCOMPUTER
REQUIREMENTS

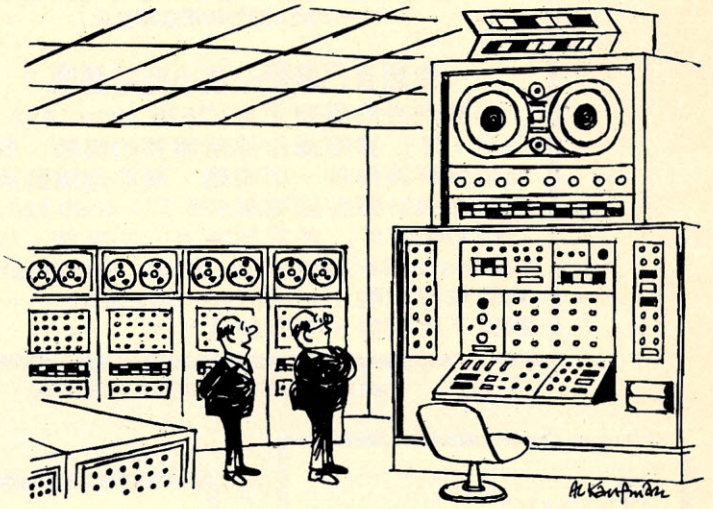
Frazer Mall, Rtes. 30 & 352
Malvern, PA 19355
Phone: (215) 647-8463

CIRCLE 143 ON READER SERVICE CARD

THE LIGHTER SIDE OF COMPUTER HARDWARE

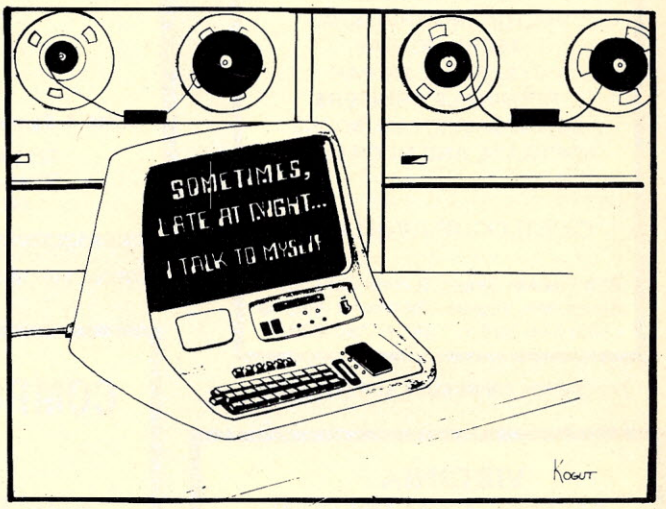


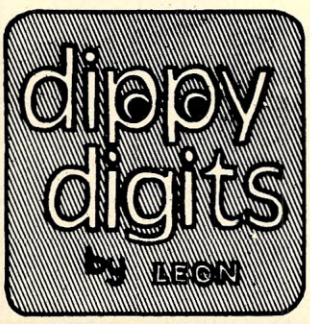
"Good grief! It's sprung a leak!!"

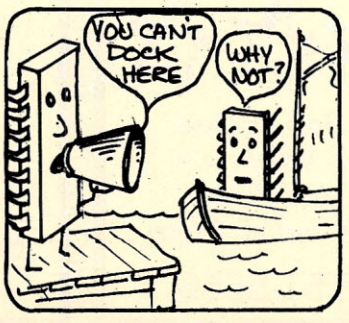


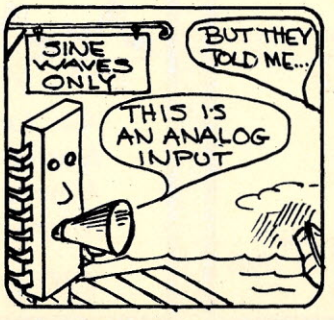
"It can't actually think, but when it makes a mistake, it puts the blame on some other computer."

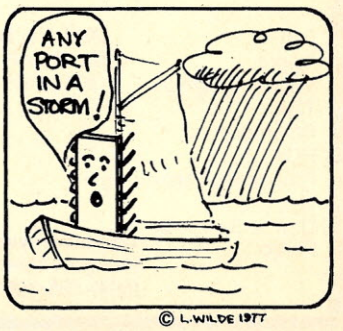
"You forgot to tell him it's an antique computer ... he thinks it's ours..."

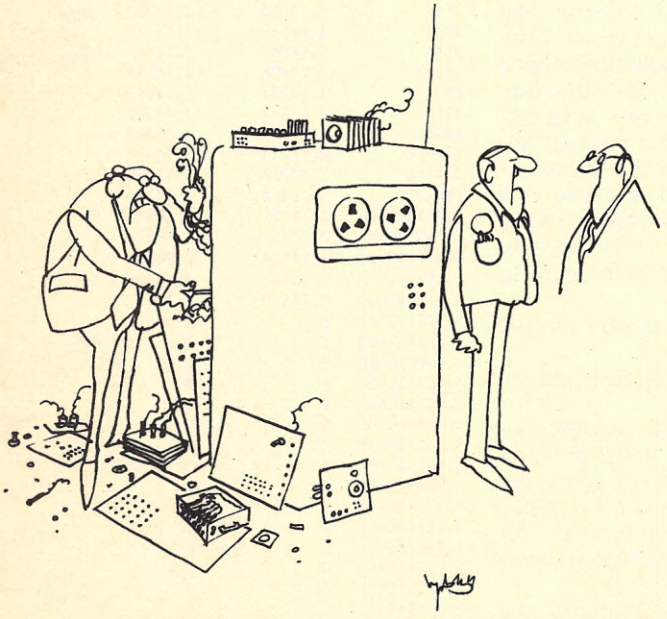












"He said he could fix it up in half the time it was taking me, so I let him try..."

"...We don't have your system designed yet but I brought along what we have..."

short programs.....

Convergence

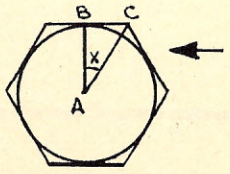
Certain constants such as e and can be calculated as the sum of a number of elements in a series. However, if such a series is calculated by hand, or even with a small calculator, it is often difficult to carry it to many digits of accuracy. The computer can easily calculate such series, although to carry the calculations to more significant digits than your compiler or interpreter permits, you'll have to use a modification of the program presented here.

Convergence on e by an infinite series. The logarithmic constant e can be represented by the series:
1 + 1/1 + 1/2 + 1/6 + 1/24 + 1/120 + . . . If you decide to extend your precision, the value of e to 15 places is 2.718281828459045.

```
LIST

10 REM:  CONVERGE ON E
20 D=1
30 E=1
40 I=0
50 I=I+1
60 D=D*I
70 E=E+1/D
80 PRINT I,E
90 GOTO 50
100 END
```

```
RUN
 1            2
 2            2.5
 3            2.66667
 4            2.70833
 5            2.71667
 6            2.71806
 7            2.71825
 8            2.71828
 9            2.71828
 10           2.71828
 11           2.71828
 12           2.71828
 13           2.71828
 14           2.71828
 15           2.71828
 16           2.71828
```



Convergence on π by infinite series. The series to converge on π is:
1 - 1/3 + 1/5 -1/7 + 1/9 . . .
Since the series converges extremely slowly, only every 500th value is printed out in the program.

```
LIST

10 P=0
20 S=1
30 I=1
40 FOR J=1 TO 499
50 P=P+S/I
60 I=I+2
70 S=-S
80 NEXT J
90 PRINT P*4
100 GOTO 40
110 END
```

```
RUN
3.1436
3.14059
3.14226
3.14109
3.142
3.14126
3.14188
3.14135
3.14182
3.1414
3.14178
3.14144
3.14176
3.14146
3.14174
3.14148
3.14172
3.14149
3.14171
3.1415
3.1417
3.14151
3.14169
```

Convergence on π by polygons. Does a square look like a circle? Not really. How about an octagon? Well, more so. A 100-sided polygon? That's getting close, but how close? One way of determining "how close?" is to calculate the perimeter of a polygon inscribed inside a circle and another circumscribed outside of a circle and when the two get very close, you've practically got a circle. This is also an interesting, if not very efficient way, to calculate π.

Consider a polygon circumscribed around a circle of radius 1.
Perimeter = length of side x no. of sides

Since the tangent of X = AB/BC, but BC = 1, then tan (X) = AB and the length of a side = 2x tan (X). Since the circumference = 2 π r and r = 1, then π is simply the circumference (or perimeter of a n-sided polygon) divided by 2.

Similar trigonometry leads to the perimeter of an inscribed polygon being equal to no. of sides x sin (X) x cos (X).

Unfortunately, there is one large fallacy in the program in that degrees must be converted into radians in statement 30. This means, of course, that you must already have the conversion factor, which is simply 2 π.

Does anyone want to guess what happened to this poor program after we got above a 768-sided polygon. What happened when we hit 50331700 sides?

```
LIST

10 N=6
20 N=2*N
30 X=6.2831853/N
40 PRINT N/2,N*SIN(X)*COS(X)/2,N*TAN(X)/2
50 GOTO 20
60 END
OK
```

```
RUN
 6            2.59808       3.4641
 12           3             3.21539
 24           3.10583       3.15966
 48           3.13263       3.14609
 96           3.13935       3.14271
 192          3.14104       3.14188
 384          3.14144       3.14165
 768          3.14158       3.14163
 1536         3.14154       3.14155   ←
 3072         3.14169       3.14169   What's
 6144         3.1414        3.1414    happening?
 12288        3.14198       3.14198
 24576        3.14083       3.14083
 49152        3.14313       3.14313
 98304        3.13853       3.13853
 196608       3.14773       3.14773
 393216       3.12932       3.12932
 786432       3.16614       3.16614
 1.57286E+06  3.09251       3.09251
 3.14573E+06  3.23977       3.23977
 6.29146E+06  2.94524       2.94524
 1.25829E+07  3.53429       3.53429
 2.51658E+07  2.35619       2.35619
 5.03317E+07  0             0
 1.00663E+08  0             0         Gulp!
 2.01327E+08  0             0         ☹
 4.02653E+08  0             0
 8.05306E+08  0             0
 1.61061E+09  0             0
 3.22123E+09  0             0
 6.44245E+09  0             0
 1.28849E+10  0             0
 2.57698E+10  0             0
 5.15396E+10  0             0
 1.03079E+11  0             0
 2.06158E+11  0             0
 4.12317E+11  0             0
 8.24634E+11  0             0
 1.64927E+12  0             0
 3.29854E+12  0             0
 6.59707E+12  0             0
 1.31941E+13  0             0
```

OIL COMPANY

Mark Phaneuf

This program puts the player in charge of a petroleum company. Through investment, your profits are increased or decreased. The program, written in DEC BASIC-OLUS for a PDP-11/50, is lengthy but is easily run and challenging to win.

Some notes on the BASIC-PLUS functions:

If no verb is stated before the statement, LET is assumed.

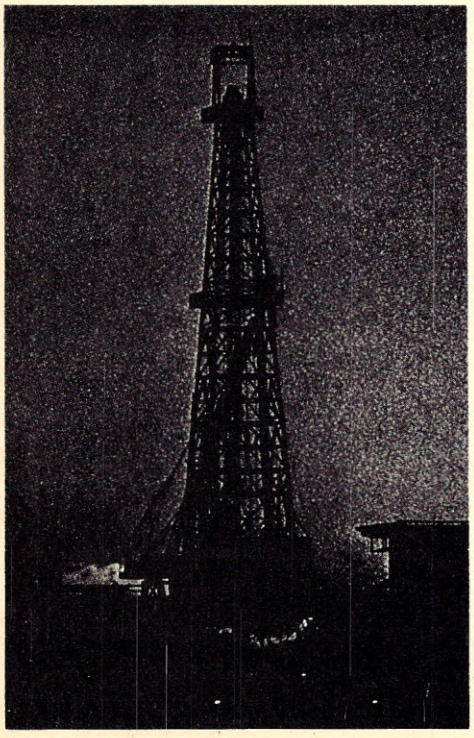
Multiple statements are used; there is more than one statement per line.

A formating procedure is used to print multiple statements. It prints the first statement beside the line number and the remaining statements below, as in Line 190.

The "&" = Print.

Statements such as in Line 280 are print-using statements. Line 280 will print "P" inserting commas "," and a decimal point "." and rounding off.

Mark Phaneuf, 81 Tulsa St., Springfield, MA 01118.



Sample Run

```
YOU HAVE JUST INHERITED AN OIL COMPANY. THIS OIL COMPANY WAS
LEFT TO YOU IN A WILL OF A DISTANT UNCLE. HE WISHED THAT
YOU WOULD INVEST HIS MONEY WISELY SO THAT YOU COULD KEEP
THE OIL COMPANY IN THE FAMILY. DON'T EISAPPOINT YOUR UNCLE
BY MAKING BAD INVESTMENTS
DO YOU WANT INSTRUCTIONS? N
HOW MANY YEARS DO YOU WANT TO PLAY (5-100)? 25
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 1
          ASSETS       $50,000.00
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       2,000.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 1
COMMAND:4
REFINING COMMAND:2
          GASOLINE             $0.76
          FUEL OIL             $0.49
          OIL PRODUCTS         $0.61
          KEROSENE             $0.62
          LUBRICANTS           $0.86
REFINING COMMAND:1
WHAT DO YOU WANT TO REFINE? F
THE REFINING PRICE OF FUEL OIL IS  $0.49
HOW MUCH FUEL OIL TO REFINE? 2000
REFINING COMMAND:4
ROYALTIES COME IN!!!!
YOU RECEIVE          726.00 DOLLARS
YOU RECEIVE          871.00 GALLONS OF CRUDE OIL
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 2
          ASSETS       $49,749.60
          TOTAL NUMBER OF GALLONS OF CRUDE OIL         871.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 1
COMMAND:1
 THE COST OF IMPORTED OIL IS  $19.85 FOR 50 GALLONS
HOW MANY BARRELS DO YOU WANT TO BUY? 100
ROYALTIES COME IN!!!!
YOU RECEIVE          602.00 DOLLARS
YOU RECEIVE          974.00 GALLONS OF CRUDE OIL
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 3
          ASSETS       $48,366.40
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       6,845.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 1
COMMAND:2
EXPORTING COMMAND:2
THE PRODUCTS IN STOCK ARE:
          GASOLINE                       0.00 GALLONS
          FUEL OIL                   1,940.00 GALLONS
          OIL PRODUCTS                   0.00 GALLONS
          KEROSENE                       0.00 GALLONS
          LUBRICANTS                     0.00 GALLONS
EXPORTING COMMAND:3
          GASOLINE             $1.09
          FUEL OIL             $0.42
          OIL PRODUCTS         $0.72
          KEROSENE             $0.85
          LUBRICANTS           $0.70
EXPORTING COMMAND:3
          GASOLINE             $1.09
          FUEL OIL             $0.42
          OIL PRODUCTS         $0.72
          KEROSENE             $0.85
          LUBRICANTS           $0.70
EXPORTING COMMAND:1
WHAT DO YOU WANT TO EXPORT? F
THE EXPORTING PRICE OF FUEL OIL IS   $0.42
HOW MUCH FUEL OIL TO EXPORT? 1900
EXPORTING COMMAND:4
ROYALTIES COME IN!!!!
YOU RECEIVE          923.00 DOLLARS
YOU RECEIVE          722.00 GALLONS OF CRUDE OIL
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 4
          ASSETS       $50,095.00
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       7,567.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 1
COMMAND:3
THE PRICE OF A REFINERY IS  $31,244.80
DO YOU WANT TO BUILD A REFINERY? YES
HOW MANY REFINERIES TO BUILD? 1
STORAGE FIRE! HALF OF ALL PRODUCTS IN STORAGE ARE DESTROYED
ROYALTIES COME IN!!!!
YOU RECEIVE          554.00 DOLLARS
YOU RECEIVE          499.00 GALLONS OF CRUDE OIL
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
```

```
                    YEAR 5
          ASSETS       $19,404.20
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       8,066.00
          TOTAL # OF REFINERIES 2
          TOTAL # OF WELLS OWNED 1
COMMAND:7
WHERE DO YOU WANT TO DRILL? 2,2
NO OIL FOUND
DO YOU WANT TO DRILL AGAIN? Y
WHERE DO YOU WANT TO DRILL? 3,4
NO OIL FOUND
DO YOU WANT TO DRILL AGAIN? 6,7
DO YOU WANT TO DRILL AGAIN? Y
WHERE DO YOU WANT TO DRILL? 6,7
OIL FOUND AT 6,000 FEET
DO YOU WANT TO DRILL AGAIN? Y
WHERE DO YOU WANT TO DRILL? 8,1
OIL FOUND AT 2,000 FEET
YOU RAN OUT OF MONEY
ROYALTIES COME IN!!!!
YOU RECEIVE          1,208.00 DOLLARS
YOU RECEIVE          1,336.00 GALLONS OF CRUDE OIL
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 6
          ASSETS       $2,612.18
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       9,402.00
          TOTAL # OF REFINERIES 2
          TOTAL # OF WELLS OWNED 2
COMMAND:4
REFINING COMMAND:2
          GASOLINE             $0.50
          FUEL OIL             $0.56
          OIL PRODUCTS         $0.61
          KEROSENE             $0.65
          LUBRICANTS           $0.83
REFINING COMMAND:1
WHAT DO YOU WANT TO REFINE? G
THE REFINING PRICE OF GASOLINE IS  $0.50
HOW MUCH GASOLINE TO REFINE? 4000
REFINING COMMAND:4
ROYALTIES COME IN!!!!
YOU RECEIVE          1,702.00 DOLLARS
YOU RECEIVE          1,416.00 GALLONS OF CRUDE OIL
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 7
          ASSETS       $2,297.45
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       6,818.00
          TOTAL # OF REFINERIES 2
          TOTAL # OF WELLS OWNED 2
COMMAND:5
          GASOLINE                  3,880
          FUEL OIL                     20
          OIL PRODUCTS                  0
          KEROSENE                      0
          LUBRICANTS                    0
COMMAND:6
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 2  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 3  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 6  0  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0
 7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
COMMAND:8
DO YOU WANT TO SELL REFINERIES OR WELLS? R
 THE SELLING PRICE OF A REFINERY IS  $20,038.60
HOW MANY REFINERIES TO SELL? 1
ROYALTIES COME IN!!!!
YOU RECEIVE          1,670.00 DOLLARS
YOU RECEIVE            748.00 GALLONS OF CRUDE OIL
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 8
          ASSETS       $24,006.00
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       7,566.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 2
COMMAND:11
ROYALTIES COME IN!!!!
YOU RECEIVE            656.00 DOLLARS
YOU RECEIVE          1,580.00 GALLONS OF CRUDE OIL
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 9
          ASSETS       $24,662.00
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       9,146.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 2
COMMAND:5
          GASOLINE                  3,880
          FUEL OIL                     20
          OIL PRODUCTS                  0
          KEROSENE                      0
          LUBRICANTS                    0
COMMAND:9
<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>
                    YEAR 9
          ASSETS       $24,662.00
          TOTAL NUMBER OF GALLONS OF CRUDE OIL       9,146.00
          TOTAL # OF REFINERIES 1
          TOTAL # OF WELLS OWNED 2
COMMAND:10
SURE?????? YES
++++ SOLD OUT FOR      50,716.60 DOLLARS ++++
********************* OWNING CERTIFICATE **********************
YEARS LASTED 9      WELLS OWNED  2
ASSETS                    $75,378.60 DOLLARS
CRUDE OIL                   9,146.00 GALLONS
REFINERIES                      1.00
GASOLINE                    3,880.00 GALLONS
FUEL OIL                      20.00 GALLONS
OIL PRODUCTS                   0.00 GALLONS
KEROSENE                       0.00 GALLONS
LUBRICANTS                     0.00 GALLONS
RATING (1 BAD) - (10 PERFECT) 0
```

Program Listing

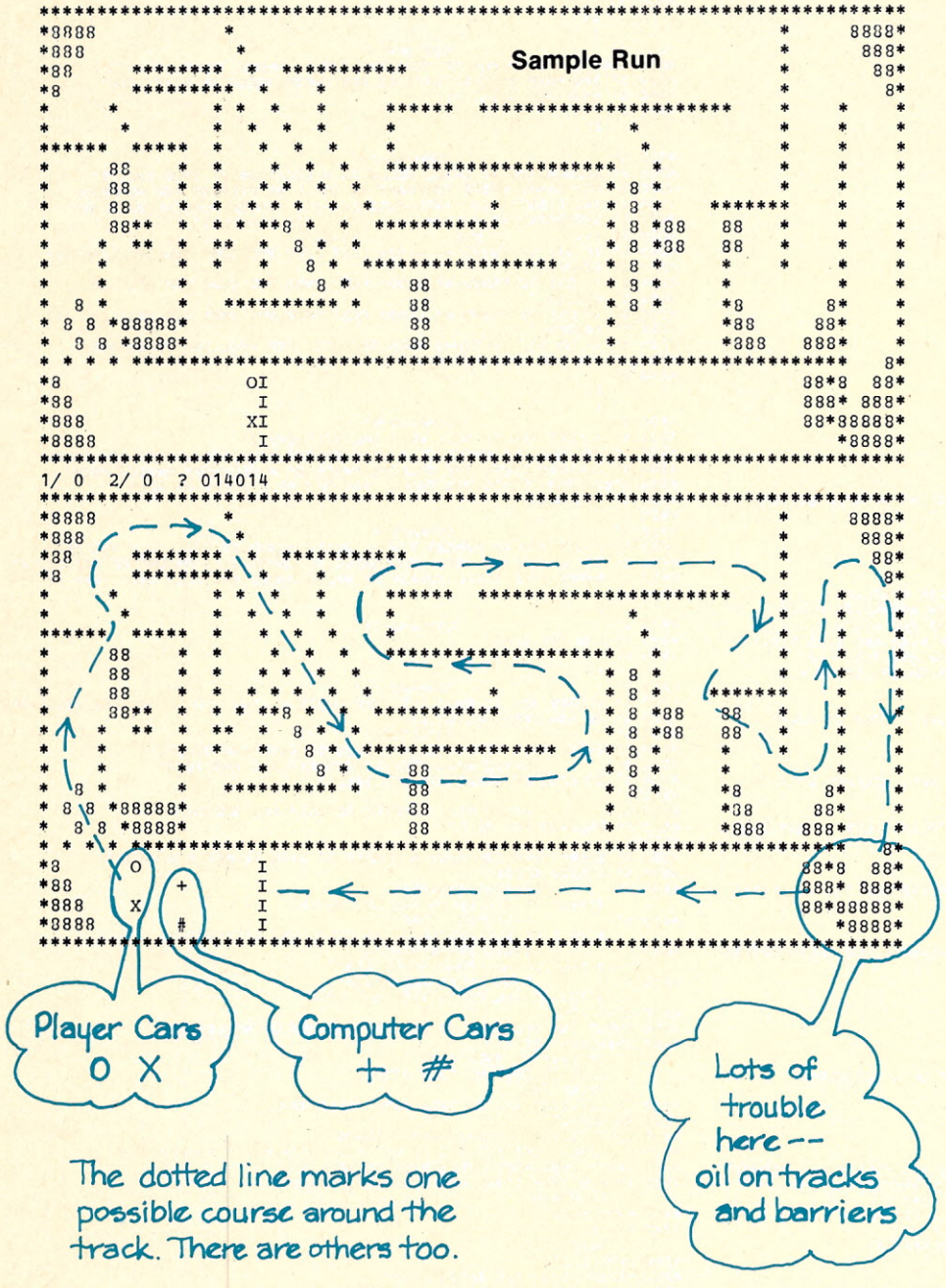
```
1 ! SAVE, KNGOIL.BAS, PHANEUF, KIM, KIM1
10 ON ERROR GOTO 4850
20 OPEN 'KB:' AS FILE 1
30 $"YOU HAVE JUST INHERITED AN OIL COMPANY. THIS OIL COMPANY WAS"
40 $"LEFT TO YOU IN A WILL OF A DISTANT UNCLE. HE WISHED THAT "
50 $"YOU WOULD INVEST HIS MONEY WISELY SO THAT YOU COULD KEEP"
60 $"THE OIL COMPANY IN THE FAMILY. DON'T EISAPPOINT YOUR UNCLE"
70 $"BY MAKING BAD INVESTMENTS"
80 DIM Y%(10,20),Z%(10,20)
90 RANDOMIZE
100 Y1=0
110 FOR I=0 TO 10
120 FOR J=0 TO 20
130 X=INT(5-1)*RND(1)+1
140 Y%(I,J)=X
150 NEXT J
160 NEXT I
170 Y1=Y1+1
180 IF Y1=200 THEN 190  ELSE 130
190 D=1:
        P=50000:
        C=2000:
        R=1:
        G=0:
        F=0:
        K=0:
        L=0:
        O=0:
        W=1:
        X1=0
200 INPUT 'DO YOU WANT INSTRUCTIONS'$I$
210 IF LEFT(I$,1)='Y' THEN 3390
220 INPUT 'HOW MANY YEARS DO YOU WANT TO PLAY (5-100)'$B5
230 B5=INT(B5)
240 IF B5<5 OR B5>100 THEN 220
250 IF P<=0 THEN 3050
260 $"<*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*> <*>"
270 $"                    YEAR"$D
280 $TAB(10)$:
        % USING 'ASSETS $$$$$,$$$,$$$.$$',P
290 $TAB(10)$:
        % USING 'TOTAL NUMBER OF GALLONS OF CRUDE OIL $$$,$$$,$$$.$$',C
300 $TAB(10)$:
        $"TOTAL # OF REFINERIES"$R
310 $TAB(10)$:
        $"TOTAL # OF WELLS OWNED"$W
320 INPUT #1,'COMMAND:'$Z
330 IF Z<1 OR Z>11 OR Z<>INT(Z) THEN 340  ELSE 400
340 $"COMMANDS ARE:"
350 $"1-IMPORT OIL,2-EXPORT OIL PRODUCTS,3-BUILD REFINERIES"
360 $"4-REFINE OIL,5-STORAGE REPORT,6- OIL FIELD DATA"
370 $"7-DRILL WELLS, 8-SELL WELLS OR REFINERIES, 9-OWNERS STATUS, 10-SELL OUT
380 $"11-PASS YEAR"
390 GOTO 320
400 ON Z GOTO 410,500,1160,1320,2280,2340,2380,4650,4830,3000,4840
410 RANDOMIZE
420 B=INT(20-10)*RND(1)+10
430 % USING ' THE COST OF IMPORTED OIL IS $$$$.$$ FOR 50 GALLONS',B
440 INPUT 'HOW MANY BARRELS DO YOU WANT TO BUY'$F7
450 F7=INT(F7)
460 IF C<0 THEN 320
470 IF F7*B>P THEN 480  ELSE 490
480 % USING 'YOU ONLY HAVE   $$$,$$$,$$$.$$ DOLLARS',P:
        GOTO 440
490 C=C+(F7*50):
        P=P-(F7*B):
        GOTO 2720
500 RANDOMIZE
510 G1=(1.25-.38)*RND(1)+.38
520 F1=(1-.33)*RND(1)+.33
530 O1=(.95-.48)*RND(1)+.48
540 K1=(.95-.48)*RND(1)+.48
550 L1=(1.25-.58)*RND(1)+.58
560 INPUT #1,'EXPORTING COMMAND:'$E
570 IF E<1 OR E>4 OR E<>INT(E) THEN 580  ELSE 620
580 $"EXPORTING COMMANDS ARE:"
590 $"1-EXPORT OIL PRODUCTS,2-PRODUCTS IN STOCK"
600 $"3-EXPORTING PRICE LIST, 4-DONE EXPORTING"
610 GOTO 560
620 ON E GOTO 630,1020,1090,1150
630 INPUT 'WHAT DO YOU WANT TO EXPORT'$E$
640 IF LEFT(E$,1)='G' THEN 650  ELSE 710
650 % USING 'THE EXPORTING PRICE OF GASOLINE IS $$$.$$',G1
660 INPUT 'HOW MUCH GASOLINE TO EXPORT'$G2
670 IF G2<=0 THEN 560
680 IF G2>G THEN 690  ELSE 700
690 % USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF GASOLINE',G:
        GOTO 660
700 G=G-G2:
        P=P+(G2*G1):
        GOTO 560
710 IF LEFT(E$,1)='F' THEN 720  ELSE 780
720 % USING 'THE EXPORTING PRICE OF FUEL OIL IS  $$$.$$',F1
730 INPUT 'HOW MUCH FUEL OIL TO EXPORT'$F2
740 IF F2<=0 THEN 560
750 IF F2>F THEN 760  ELSE 770
760 % USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF FUEL OIL',F:
        GOTO 730
770 F=F-F2:
        P=P+(F2*F1):
        GOTO 560
780 IF LEFT(E$,1)='O' THEN 790  ELSE 850
790 % USING 'THE EXPORTING PRICE OF OIL PRODUCTS IS $$$.$$',O1
800 INPUT 'HOW MUCH OIL PRODUCTS TO EXPORT'$O2
810 IF O2<=0 THEN 560
820 IF O2>O THEN 830  ELSE 840
830 % USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF OIL PRODUCTS',O:
        GOTO 800
840 O=O-O2:
        P=P+(O2*O1):
        GOTO 560
850 IF LEFT(E$,1)='K' THEN 860  ELSE 920
860 % USING 'THE EXPORTING PRICE OF KEROSENE IS $$$.$$',K1
870 INPUT 'HOW MUCH KEROSENE TO EXPORT'$K2
880 IF K2<=0 THEN 560
890 IF K2>K THEN 900  ELSE 910
900 % USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF KEROSENE',K:
        GOTO 870
910 K=K-K2:
        P=P+(K2*K1):
        GOTO 560
920 IF LEFT(E$,1)='L' THEN 930  ELSE 1000
930 % USING 'THE EXPORTING PRICE OF LUBRICANTS IS $$$.$$',L1
940 INPUT 'HOW MUCH LUBRICANTS TO EXPORT'$L2
950 IF L2<=0 THEN 560
960 IF L2>L THEN 970  ELSE 980
970 % USING 'YOU ONLY HAVE $$$,$$$,$$$.$$ GALLONS OF LUBRICANTS',L:
        GOTO 940
980 L=L-L2:
        P=P+(L2*L1):
        GOTO 560
990 $"THE PRODUCTS YOU CAN EXPORT ARE:"
1000 $"GASOLINE,FUEL OIL, OIL PRODUCTS , KEROSENE, AND LUBRICANTS"
1010 GOTO 560
1020 $"THE PRODUCTS IN STOCK ARE:"
1030 $TAB(10)$:
        % USING 'GASOLINE              $$$,$$$,$$$.$$ GALLONS',G
1040 $TAB(10)$:
        % USING 'FUEL OIL              $$$,$$$,$$$.$$ GALLONS',F
1050 $TAB(10)$:
        % USING 'OIL PRODUCTS          $$$,$$$,$$$.$$ GALLONS',O
1060 $TAB(10)$:
        % USING 'KEROSENE              $$$,$$$,$$$.$$ GALLONS',K
1070 $TAB(10)$:
        % USING 'LUBRICANTS            $$$,$$$,$$$.$$ GALLONS',L
1080 GOTO 560
1090 $TAB(10)$:
        % USING 'GASOLINE              $$$.$$',G1
1100 $TAB(10)$:
        % USING 'FUEL OIL              $$$.$$',F1
1110 $TAB(10)$:
        % USING 'OIL PRODUCTS          $$$.$$',O1
1120 $TAB(10)$:
        % USING 'KEROSENE              $$$.$$',K1
1130 $TAB(10)$:
        % USING 'LUBRICANTS            $$$.$$',L1
1140 GOTO 560
1150 GOTO 2720
1160 RANDOMIZE
1170 R1=INT(50000-30000)*RND(1)+30000
1180 % USING 'THE PRICE OF A REFINERY IS $$$$,$$$.$$',R1
1190 INPUT 'DO YOU WANT TO BUILD A REFINERY'$R$
1200 IF LEFT(R$,1)='Y' THEN 1240  ELSE 1210
1210 IF LEFT(R$,1)='N' THEN 320  ELSE 1220
1220 $"  I NEED (YES OR NO)"
1230 GOTO 1190
1240 INPUT 'HOW MANY REFINERIES TO BUILD'$R2
1250 IF R2<0 OR R2<>INT(R2) THEN 1260  ELSE 1280
1260 $"YOU HAVE TO BUY EITHER 0 OR ANY WHOLE NUMBER OF REFINERIES"
1270 GOTO 1240
1280 IF R2*R1>P THEN 1290  ELSE 1310
1290 % USING 'YOU ONLY HAVE    $$$,$$$,$$$.$$ DOLLARS',P
1300 GOTO 1240
1310 P=P-(R2*R1):
        R=R+R2:
        GOTO 2720
1320 RANDOMIZE
1330 G3=(1-.5)*RND(1)+.5
1340 F3=(.75-.45)*RND(1)+.45
1350 O3=(.7-.6)*RND(1)+.6
1360 L3=(1-.7)*RND(1)+.7
1370 K3=(.7-.6)*RND(1)+.6
1380 INPUT #1,'REFINING COMMAND:'$R3
1390 IF R3<1 OR R3>4 OR R3<>INT(R3) THEN 1400  ELSE 1430
1400 $"THE REFINING COMMANDS ARE:"
1410 $"1-REFINE,2-REFINING PRICE LIST,3-COMPANY STATUS,4-DONE REFINING"
1420 GOTO 1380
1430 ON R3 GOTO 1440,2180,2240,2270
1440 INPUT 'WHAT DO YOU WANT TO REFINE'$A$
1450 IF LEFT(A$,1)='G' THEN 1460  ELSE 1590
1460 % USING 'THE REFINING PRICE OF GASOLINE IS $$$.$$',G3
1470 INPUT 'HOW MUCH GASOLINE TO REFINE'$G4
1480 G4=INT(G4)
1490 IF G4<=0 THEN 1380
1500 IF G4>C THEN 1510  ELSE 1520
1510 % USING 'YOU ONLY HAVE $$$,$$$,$$$ GALLONS OF CRUDE OIL',C:
        GOTO 1470
1520 IF G4/125000>R THEN 1530  ELSE 1540
1530 $"THAT IS TOO MUCH CRUDE OIL FOR YOUR REFINERIES":
        GOTO 1470
1540 IF G4*G3>P THEN 1550  ELSE 1560
1550 $"YOU DON'T HAVE ENOUGH MONEY":
        GOTO 1470
1560 IF (X1+G4)/125000>R THEN 1570  ELSE 1580
1570 $"YOU CAN REFINE"$(R*125000)-X1$"GALLONS":
        GOTO 1470
1580 G=G+(G4-(G4*.03)):
        G=INT(G):
        P=P-(G4*G3):
        C=C-G4:
        X1=X1+G4:
        GOTO 1380
```

```
1590 IF LEFT(A$,1)='F' THEN 1600  ELSE 1730
1600 % USING 'THE REFINING PRICE OF FUEL OIL IS $$$.$$',F3
1610 INPUT 'HOW MUCH FUEL OIL TO REFINE'#F4
1620 F4=INT(F4)
1630 IF F4<=0 THEN 1380
1640 IF F4>C THEN 1650  ELSE 1660
1650 % USING 'YOU ONLY HAVE ###,###,### GALLONS OF CRUDE OIL',C:
       GOTO 1610
1660 IF F4/125000>R THEN 1670  ELSE 1680
1670 $'THAT IS TOO MUCH CRUDE OIL FOR YOUR REFINERIES':
       GOTO 1610
1680 IF F4*F3>P THEN 1690  ELSE 1700
1690 $'YOU DON'T HAVE ENOUGH MONEY':
       GOTO 1610
1700 IF (X1+F4)/125000>R THEN 1710  ELSE 1720
1710 $'YOU CAN REFINE'#(R*125000)-X1#'GALLONS':
       GOTO 1610
1720 F=F+(F4-(F4*.03)):
       F=INT(F):
       P=P-(F4*F3):
       C=C-F4:
       X1=X1+F4:
       GOTO 1380
1730 IF LEFT(A$,1)='O' THEN 1740  ELSE 1870
1740 % USING ' THE REFINING PRICE OF OIL PRODUCTS IS $$$.$$',O3
1750 INPUT 'HOW MUCH OIL PRODUCTS TO REFINE'#O4
1760 O4=INT(O4)
1770 IF O4<=0 THEN 1380
1780 IF O4>C THEN 1790  ELSE 1800
1790 % USING 'YOU ONLY HAVE ###,###,### GALLONS OF CRUDE OIL',C:
       GOTO 1750
1800 IF O4/125000>R THEN 1810  ELSE 1820
1810 $'THAT IS TOO MUCH CRUDE OIL FOR YOUR REFINERIES':
       GOTO 1750
1820 IF O4*O3>P THEN 1830  ELSE 1840
1830 $'YOU DON'T HAVE ENOUGH MONEY':
       GOTO 1750
1840 IF (X1+O4)/125000>R THEN 1850  ELSE 1860
1850 $'YOU CAN REFINE'#(R*125000)-X1#'GALLONS':
       GOTO 1750
1860 O=O+(O4-(O4*.03)):
       O=INT(O):
       P=P-(O4*O3):
       C=C-O4:
       X1=X1+O4:
       GOTO 1380
1870 IF LEFT(A$,1)='K' THEN 1880  ELSE 2010
1880 % USING   THE REFINING PRICE OF KEROSENE IS $$$.$$',K3
1890 INPUT 'HOW MUCH KEROSENE TO REFINE'#K4
1900 K4=INT(K4)
1910 IF K4<=0 THEN 1380
1920 IF K4>C THEN 1930  ELSE 1940
1930 % USING 'YOU ONLY HAVE ###,###,### GALLONS OF CRUDE OIL',C:
       GOTO 1890
1940 IF K4/125000>R THEN 1950  ELSE 1960
1950 $'THAT IS TOO MUCH CRUDE OIL FOR YOUR REFINERIES':
       GOTO 1890
1960 IF K4*K3>P THEN 1970  ELSE 1980
1970 $'YOU DON'T HAVE ENOUGH MONEY':
       GOTO 1890
1980 IF (X1/K4)/125000>R THEN 1990  ELSE 2000
1990 $'YOU CAN REFINE'#(R*125000)-X1#'GALLONS':
       GOTO 1890
2000 K=K+(K4-(K4*.03)):
       K=INT(K):
       P=P-(K4*K3):
       C=C-K4:
       X1=X1+K4:
       GOTO 1380
2010 IF LEFT(A$,1)='L' THEN 2020  ELSE 2150
2020 % USING ' THE REFINING PRICE OF LUBRICANTS IS $$$.$$',L3
2030 INPUT 'HOW MUCH LUBRICANTS TO REFINE'#L4
2040 L4=INT(L4)
2050 IF L4<=0 THEN 1380
2060 IF L4>C THEN 2070  ELSE 2080
2070 % USING 'YOU ONLY HAVE ###,###,### GALLONS OF CRUDE OIL',C:
       GOTO 2030
2080 IF L4/125000>R THEN 2090  ELSE 2100
2090 $'THAT IS TOO MUCH FOR YOUR REFINERIES ':
       GOTO 2030
2100 IF L4*L3>P THEN 2110  ELSE 2120
2110 $'YOU DON'T HAVE ENOUGH MONEY':
       GOTO 2030
2120 IF (X1+L4)/125000>R THEN 2130  ELSE 2140
2130 $'YOU CAN REFINE'#(R*125000)-X1#'GALLONS':
       GOTO 2030
2140 L=L+(L4-(L4*.03)):
       L=INT(L):
       P=P-(L4*L3):
       C=C-L4:
       X1=X1+L4:
       GOTO 1380
2150 $'THE PRODUCTS THAT YOU CAN REFINE ARE:'
2160 $'GASOLINE, FUEL OIL, OIL PRODUCTS, KEROSENE, AND LUBRICANTS'
2170 GOTO 1440
2180 $TAB(10)#:
       % USING 'GASOLINE            $$.$$',G3
2190 $TAB(10)#:
       % USING 'FUEL OIL            $$.$$',F3
2200 $TAB(10)#:
       % USING 'OIL PRODUCTS        $$.$$',O3
2210 $TAB(10)#:
       % USING 'KEROSENE            $$.$$',K3
2220 $TAB(10)#:
       % USING 'LUBRICANTS          $$.$$',L3
2230 GOTO 1380
2240 % USING 'ASSETS                $$$,$$$,$$$.$$ DOLLARS',P
2250 % USING 'CRUDE OIL            ###,###,###.## GALLONS',C
2260 GOTO 1380
2270 GOTO 2720
2280 $TAB(10)#:
       % USING 'GASOLINE            $$$,$$$,$$$',G
2290 $TAB(10)#:
       % USING 'FUEL OIL            $$$,$$$,$$$',F
2300 $TAB(10)#:
       % USING 'OIL PRODUCTS        $$$,$$$,$$$',O
2310 $TAB(10)#:
       % USING 'KEROSENE            $$$,$$$,$$$',K
2320 $TAB(10)#:
       % USING 'LUBRICANTS          $$$,$$$,$$$',L
2330 GOTO 320
2340 % USING '## ',J# FOR J=0 TO 20:
       $:
       FOR I=1 TO 10
2350 % USING '## ',I#:
       % USING '## ',Z%(I,J)# FOR J=1 TO 20
2360 $:
       NEXT I
2370 GOTO 320
2380 INPUT 'WHERE DO YOU WANT TO DRILL'#I,J
2390 IF I>10 OR J>20 OR I<1 OR J<1 THEN 2400  ELSE 2410
2400 $'I NEED 2 WHOLE NUMBERS THE 1ST(1-10) THEN 2ND(1-20)':
       GOTO 2380
2410 I=INT(I):
       J=INT(J):
       IF W9=200 THEN 2990
2420 ON Y%(I,J) GOTO 2590,2540,2490,2440,2430
2430 $'YOU ALREADY DRILLED THERE':
       GOTO 2380
2440 $'OIL FOUND AT 2,000 FEET':
       P=P-2000
2450 IF P<0 THEN 2460  ELSE 2470
2460 P=P+2000:
       $'YOU RAN OUT OF MONEY':
       GOTO 2720
2470 Y%(I,J)=5
2480 Z%(I,J)=5:
       GOTO 2640
2490 $'OIL FOUND AT 4,000 FEET':
       P=P-4000
2500 IF P<0 THEN 2510  ELSE 2520
2510 P=P+4000:
       $'YOU RAN OUT OF MONEY':
       GOTO 2720
2520 Y%(I,J)=5
2530 Z%(I,J)=5:
       GOTO 2640
2540 $'OIL FOUND AT 6,000 FEET':
       P=P-6000
2550 IF P<0 THEN 2560  ELSE 2570
2560 P=P+6000:
       $'YOU RAN OUT OF MONEY':
       GOTO 2720
2570 Y%(I,J)=5
2580 Z%(I,J)=5:
       GOTO 2640
2590 $'NO OIL FOUND':
       P=P-6000
2600 IF P<0 THEN 2610  ELSE 2620
2610 P=P+6000:
       $'YOU RAN OUT OF MONEY':
       GOTO 2720
2620 Y%(I,J)=5
2630 Z%(I,J)=5:
       GOTO 2650
2640 W=W+1
2650 W9=W9+1
2660 INPUT 'DO YOU WANT TO DRILL AGAIN'#D$
2670 IF D$='Y' OR D$='N' THEN 2680  ELSE 2660
2680 IF LEFT(D$,1)='Y' THEN 2380
2690 GOTO 2720
2700 IF LEFT(Y9$,1)='1' THEN 2720  ELSE 2710
2710 IF LEFT(Y9$,1)='2' THEN 320  ELSE 2690
2720 RANDOMIZE
2730 U=INT(60*RND(1))
2740 IF U=50 THEN 2750  ELSE 2760
2750 R=R*.75:
       R=INT(R):
       GOTO 2780
2760 IF U=40 THEN 2770  ELSE 2810
2770 P=P*.5:
       P=INT(P):
       GOTO 2790
2780 $'REFINERIES EXPLODE'#R#'REFINERIES LEFT':
       GOTO 2890
2790 % USING 'BAD INVESTMENT   ###,###,###.## DOLLARS LEFT',P
2800 GOTO 2890
2810 IF U=30 THEN 2820  ELSE 2840
2820 % USING 'OIL SPILL! ###,###,### GALLONS OF CRUDE OIL DESTROYED',C*.75
2830 C=C-(C*.75):
       GOTO 2890
2840 IF U=20 THEN 2850  ELSE 2860
2850 $'FIRE DAMAGE!'#W*.5#'WELLS DESTROYED':
       W=W-(W*.5):
       GOTO 2890
2860 IF U=10 THEN 2870  ELSE 2890
2870 $'STORAGE FIRE! HALF OF ALL PRODUCTS IN STORAGE ARE DESTROYED'
2880 G=G/2:
       F=F/2:
       L=L/2:
       K=K/2:
       O=O/2
2890 D=D+1:
       X1=0
2900 IF W>0 THEN $'ROYALTIES COME IN!!!!'  ELSE 2970
2910 RANDOMIZE
2920 W1=INT(1000-100)*RND(1)+100:
       C5=INT(1000-100)*RND(1)+100
2930 W1=INT(W1):
       C5=INT(C5)
2940 P=P+(W*W1):
       C=C+(C5*W)
2950 % USING 'YOU RECEIVE   ###,###,###.## DOLLARS',W*W1
2960 % USING 'YOU RECEIVE   ###,###,###.## GALLONS OF CRUDE OIL',C5*W
2970 IF W9=200 OR D=B5 OR P=0 THEN 2990  ELSE 2980
2980 G=INT(G):
       F=INT(F):
       O=INT(O):
       K=INT(K):
       L=INT(L):
       W=INT(W):
       GOTO 250
```

```
2990 &"GAME OVER":
     GOTO 3050
3000 INPUT 'SURE?????';M9$
3010 IF M9$='YES' THEN 3030
3020 IF M9$='NO' THEN 320
     ELSE &" YES OR NO":
     GOTO 3000
3030 % USING '++++ SOLD OUT FOR $$$,$$$,$$$.$$ DOLLARS ++++',((W+P+C+R)*1.
3040 P=P+((W+P+C+R)*1.5)
3050 &"********************* OWNING CERTIFICATE *********************"
3060 &"YEARS LASTED";D;"    WELLS OWNED ";W
3070 % USING 'ASSETS                $$$$$,$$$,$$$.$$ DOLLARS',P
3080 % USING 'CRUDE OIL               $$$,$$$,$$$.$$ GALLONS',C
3090 % USING 'REFINERIES              $$$,$$$,$$$.$$',R
3100 % USING 'GASOLINE                $$$,$$$,$$$.$$ GALLONS',G
3110 % USING 'FUEL OIL                $$$,$$$,$$$.$$ GALLONS',F
3120 % USING 'OIL PRODUCTS            $$$,$$$,$$$.$$ GALLONS',O
3130 % USING 'KEROSENE                $$$,$$$,$$$.$$ GALLONS',K
3140 % USING 'LUBRICANTS              $$$,$$$,$$$.$$ GALLONS',L
3150 Q3=(((P+C)/5000)+((W+R)*.1)*(D-1))
3160 IF Q3>=486.5 THEN 3170  ELSE 3180
3170 R9=10:
     GOTO 3370
3180 IF Q3>=437.85 THEN 3190  ELSE 3200
3190 R9=9:
     GOTO 3370
3200 IF Q3>=389.2 THEN 3210  ELSE 3220
3210 R9=8:
     GOTO 3370
3220 IF Q3>=340.55 THEN 3230  ELSE 3240
3230 R9=7:
     GOTO 3370
3240 IF Q3>=291.9 THEN 3250  ELSE 3260
3250 R9=6:
     GOTO 3370
3260 IF Q3>=243.25 THEN 3270  ELSE 3280
3270 R9=5:
     GOTO 3370
3280 IF Q3>=194.6 THEN 3290  ELSE 3300
3290 R9=4:
     GOTO 3370
3300 IF Q3>=145.95 THEN 3310  ELSE 3320
3310 R9=3:
     GOTO 3370
3320 IF Q3>=97.3 THEN 3330  ELSE 3340
3330 R9=2:
     GOTO 3370
3340 IF Q3>=48.65 THEN 3350  ELSE 3360
3350 R9=1:
     GOTO 3370
3360 R9=0:
     GOTO 3370
3370 &"RATING (1 BAD) - (10 PERFECT)";R9
3380 GOTO 4860
3390 &"                    COMMAND-1"
3400 &" ALLOWS YOU TO PURCHASE IMPORTED CRUDE OIL AT A PRICE BETWEEN"
3410 &"$10.00 AND $20.00 PER FIFTY GALLON BARREL"
3420 &:
     &:
     &
3430 &"                    COMMAND-2"
3440 &" ALLOWS YOU TO EXPORT YOUR REFINED PRODUCTS NAMELY;"
3450 &" GASOLINE SELLING FOR A PRICE BETWEEN $.38 AND $1.25 P.G."
3460 &" FUEL OIL SELLING FOR A PRICE BETWEEN $.33 AND $1.00 P.G."
3470 &" OIL PRODUCTS SELLING FOR A PRICE BETWEEN $.48 AND $.95 P.G."
3480 &" KEROSENE SELLING FOR A PRICE BETWEEN $.48 AND $.95 P.G."
3490 &" LUBRICANTS SELLING FOR A PRICE BETWEEN $.58 AND $1.25 P.G."
3500 &
3510 &"                    SUGGESTIONS"
3520 &" CONSIDERATION SHOULD BE GIVEN TO EXPORTING A PRODUCT WHEN"
3530 &" ITS PRICE IS HIGH"
3540 &:
     &
3550 &" WHEN ASKED FOR AN EXPORTING COMMAND YOU MUST RESPOND WITH;"
3560 &
3570 &"         1-EXPORT"
3580 &" ALLOWS YOU TO EXPORT ANY REFINED PRODUCTS IN STOCK IN ANY GIVEN"
3590 &" QUANTITY AT THE SPECIFIED PRICE"
3600 &
3610 &"         2-PRODUCTS IN STOCK"
3620 &" GIVES YOU A LISTING OF YOUR REFINED PRODUCTS AND THE QUANTITY"
3630 &" OF EACH"
3640 &
3650 &"         3-EXPORTING PRICE LIST"
3660 &" GIVES YOU A LISTING OF ALL EXPORTABLE PRODUCTS AND THEIR PRICES"
3670 &
3680 &"         4-DONE EXPORTING"
3690 &" SIGNIFIES COMPLETION OF  COMMAND 2"
3700 &:
     &:
     &
3710 &"                    COMMAND-3"
3720 &" ALLOWS YOU TO BUILD REFINERIES. THE COST OF A REFINERY"
3730 &"RANGES FROM $30,000 TO $50,000 EACH"
3740 &:
     &:
     &
3750 &"                    COMMAND-4"
3760 &" ALLOWS YOU TO REFINE YOUR CRUDE OIL INTO THE FOLLOWING PRODUCTS"
3770 &" GASOLINE COSTING $.50 TO $1.00 P.G. TO REFINE"
3780 &" FUEL OIL COSTING $.45 TO $.75 P.G. TO REFINE"
3790 &" OIL PRODUCTS COSTING $.60 TO $.70 P.G. TO REFINE"
3800 &" KEROSENE COSTING $.60 TO $.70 P.G. TO REFINE"
3810 &" LUBRICANTS COSTING $.70 TO $1.00 P.G. TO REFINE"
3820 &:
     &
3830 &" WHEN ASKED FOR A REFINING COMMAND YOU MUST RESPOND WITH;"
3840 &"         1-REFINE"
3850 &" ALLOWS YOU TO REFINE CRUDE OIL INTO BY-PRODUCTS"
3860 &
3870 &"         2-REFINING PRICE LIST"
3880 &" GIVES YOU A LISTING OF ALL REFINABLE PRODUCTS AND THEIR PRICES"
3890 &
3900 &"         3-COMPANY STATUS"
3910 &
3920 &" TELLS YOU HOW MUCH MONEY AND CRUDE OIL YOU HAVE LEFT"
3930 &
3940 &"         4-DONE REFINING"
3950 &" SIGNIFIES COMPLETION OF COMMAND 4"
3960 &:
     &
3970 &"                    STIPULATIONS"
3980 &"         A- YOU ARE UNABLE TO REFINE MORE OF ANY PRODUCT"
3990 &" THAN YOU HAVE CRUDE OIL"
4000 &
4010 &"         B- A REFINERY IS LIMITED TO PRODUCING 125,000 GALLONS"
4020 &" PER YEAR. THEREFORE YOU ARE UNABLE TO REFINE MORE CRUDE OIL"
4030 &" THAN YOUR REFINERIES ARE CAPIBLE OF REFINING"
4040 &
4050 &"         C-THE GALLONAGE TIMES THE REFINING PRICE CAN'T"
4060 &" BE GRATER THAN YOUR ASSETS"
4070 &"                    STIPULATIONS"
4080 &" WHEN REFINING 3% OF THE CRUDE OIL THAT YOU WANTED TO REFINE"
4090 &" IS LOST IN THE REFINING PROCESS"
4100 &:
     &:
     &
4110 &"                    COMMAND-5"
4120 &" GIVES A LISTING OF ALL YOUR REFINED PRODUCTS AND THE QUANTITY"
4130 &" IN WHICH THEY WERE REFINED"
4140 &:
     &:
     &
4150 &"                    COMMAND-6"
4160 &"GIVES YOU A MAP OF YOUR OIL FIELD. WHERE THERE IS A '5'"
4170 &" YOU HAVE ALREADY DRILLED AND WHERE THERE IS A '0' YOU"
4180 &" HAVE NOT DRILL YET"
4190 &:
     &:
     &
4200 &"                    COMMAND-7"
4210 &" ALLOWS YOU TO DRILL WELLS IN A LOCATION OF YOUR CHOICE"
4220 &" YOU STATE WHERE YOU WANT TO DRILL BY PICKING TWO NUMBERS"
4230 &" THE FIRST(1-10) THEN SECOND(1-20). THESE NUMBERS MUST BE"
4240 &" SEPERATED BY A COMMA.          FOR EXAMPLE:"
4250 &" 1,1          5,5          10,20"
4260 &" IF OIL IS FOUND AT 2,000 FEET THEN THE COST FOR DRILLING IS"
4270 &" $2,000."
4280 &" IF OIL IS FOUND AT 4,000 FEET THEN THE COST FOR DRILLING IS"
4290 &" $4,000."
4300 &" IF OIL IS FOUND AT 6,000 FEET THEN THE COST FOR DRILLING IS"
4310 &" $6,000."
4320 &" IF NO OIL IS FOUND THEN THE COST FOR DRILLING IS $6,000"
4330 &" BUT YOU WILL NOT RECEIVE ANY WELL"
4340 &:
     &:
     &
4350 &"                    COMMAND-8"
4360 &" ALLOWS YOU TO SELL WELL AND REFINERIES"
4370 &" THE SELLING PRICE OF A WELL RANGES FROM $500.00 TO $5,000.00"
4380 &" DOLLARS EACH. THE SELLING PRICE OF A REFINERY RANGES FROM"
4390 &"$20,000 TO $30,000 EACH.  YOU MAY SELL A FEW OR ALL"
4400 &" YOU HAVE"
4410 &
4420 &"                    COMMAND-9"
4430 &" GIVES YOU AN OWNERS STATUS WHICH MEANS IT WILL TELL YOU YOUR"
4440 &" TOTAL ASSETS, TOTAL GALLONAGE OF CRUDE OIL, AMOUNT OF REFINERIES"
4450 &" OWNED, THE TOTAL NUMBER OF WELLS OWNED AND ALSO WHAT YEAR IT IS"
4460 &:
     &
4470 &"                    COMMAND-10"
4480 &" END THE GAME"
4490 &:
     &
4491 &"                    COMMAND-11"
4492 &" ALLOWS YOU TO PASS THE YEAR WHICH ALLOWS YOU TO COLLECT ROYALTIES"
4493 &" ON YOU EXISTING WELLS"
4500 &"                    ROYALTIES"
4510 &" EVERY YEAR ROYALTIES ARE PAID ON EXISTING WELLS."
4520 &"         A-YOU WILL RECEIVE BETWEEN 100 AND 1000"
4530 &" GALLONS OF CRUDE OIL PER WELL."
4540 &
4550 &"         B-YOU WILL RECEIVE BETWEEN 100 AND 1000"
4560 &" DOLLARS PER WELL"
4570 &
4580 &" EVERY YEAR YOU HAVE A CHANCE OF 1:15 TO GET A DISASTER"
4590 &" LISTED BELOW;"
4600 &"         A-FIRE DAMAGE":
     &"HALF OF ALL WELLS ARE DESTROYED"
4610 &"         B-STORAGE FIRE":
     &"HALF OF ALL PRODUCTS  IN STOCK ARE DESTROYED"
4620 &"         C-BAD INVESTMENT":
     &"HALF OF ALL ASSETS ARE LOST"
4630 &"         D-REFINERY EXPLOSION":
     &"75% OF ALL REFINERIES ARE LOST"
4640 GOTO 220
4650 INPUT 'DO YOU WANT TO SELL REFINERIES OR WELLS';Z5$
4660 Z5$=LEFT(Z5$,1):
     IF Z5$='R' THEN 4680
     ELSE IF Z5$='W' THEN 4750
      ELSE 320
4670 IF LEFT(Z5$,1)='W' THEN 4750  ELSE 4650
4680 RANDOMIZE
4690 X3=INT(30000-20000)*RND(1)+20000
4700 % USING ' THE SELLING PRICE OF A REFINERY IS $$$$,$$$.$$',X3
4710 INPUT 'HOW MANY REFINERIES TO SELL';H
4720 IF H>R THEN &"YOU HAVE";R;"REFINERIES":
     GOTO 4700
4730 P=P+(H*X3)
4740 R=R-H:
     GOTO 2720
4750 RANDOMIZE
4760 X5=INT(5000-500)*RND(1)+500
4770 &"THE SELLING PRICE OF A REFINERY IS ";X5
4780 INPUT 'HOW MANY WELLS TO SELL';H1
4790 IF H1>W THEN &"YOU HAVE";W;"WELLS":
     GOTO 4770
4800 P=P+(X5*H1)
4810 W=W-H1:
     GOTO 2720
4820 GOTO 260
4830 GOTO 260
4840 GOTO 2720
4850 RESUME
4860 END
```

RACETRACK

Scott Bennett



This racetrack program can be tailored to fit an individual's CRT, and is a standard one that includes scrolling. Following are some suggestions which may help the user run the program quickly and efficiently.

Scott Bennett, 709 Diana Ct., Iowa City, IA 52240.

The program works ideally on a CRT with character addressing. With character addressing you do not need to reprint the entire track each turn.

A computer function that returns the cursor to the upper left-hand corner of the video screen can be used. Each line of the display matrix is then checked to see if any change has occurred. If there is no change, the cursor is moved down one line. If a change has occurred, such as a car appearing in the row, the line is reprinted. This method has proved to be very fast and one can get almost immediate results. I was working with 1200 baud.

There are three car-controlling commands: Brake/Accelerate, Gear and Direction.

The Brake/Accelerate command has a number designation. The digits 1, 2, 3, 4, 5, 6, 7, 8, 9, and 0 range from least to greatest speed. Proceeding from 1 through 5, you go from highest to lowest braking. From 6 through 0 you go from the least to the greatest acceleration.

1 2 3 4 5 6 7 8 9 0

least braking *greatest acceleration*

As you accelerate a car, the reaction is not instantaneous, but gradual. It takes a number of turns to reach a set rate. This adds more challenge to the race. While climbing to a new rate, one must use the appropriate gear for particular ranges of speed. If this is not done, one will accelerate by only 10 mph at each turn. The following is a chart of the correct speed/gear ratios:

MPH	*GEAR*
0-25	1
25-100	2
100-200	3
200+	4

A car can be steered in any one of eight directions. With the top of the CRT screen as North, the directions and their number designations follow:

NW	N	NE	
7	8	9	
W 4		6	E
1	2	3	
SW	S	SE	

A command for operating a car would appear in the form:

Accel/Brake ; Gear ; Direction

To enter your command after the question mark, remember the following: The computer wants six numbers in a row—no commas. The first three are for the first player, car "X"; the second three are for the second player, car "O". For example:

Entry: 014018

The first player has floored the accelerator. He is between 0 and 25

miles per hour and is going West. The acceleration and gear range for the second player is the same except that he is going North.

Lines 200-380 contain data the computer uses to initiate the track. The data is read in lines 480-670.

In the program matrix, D$(24,75) represents the screen of characters, 24 down and 75 across with the upper left corner of the screen as the origin. The program loops three times: once to draw a line of **** to represent walls, once to draw a line of 8888 to represent oil slicks, and once to draw a line of IIII to represent the Finish Line.

To carry this out, the reference numbers 0, -1, and -2 are used.

The number "0" tells the computer to draw a horizontal line. The two numbers following this reference number give the vertical and horizontal coordinates, respectively, of the starting point of the line. The third number following the reference number gives the horizontal coordinate of the end-point of the line.

The -1 reference number tells the computer to draw a line of characters vertically. The two numbers that follow give the vertical and horizontal coordinates of the starting-point and the third number the vertical coordinate of the end-point.

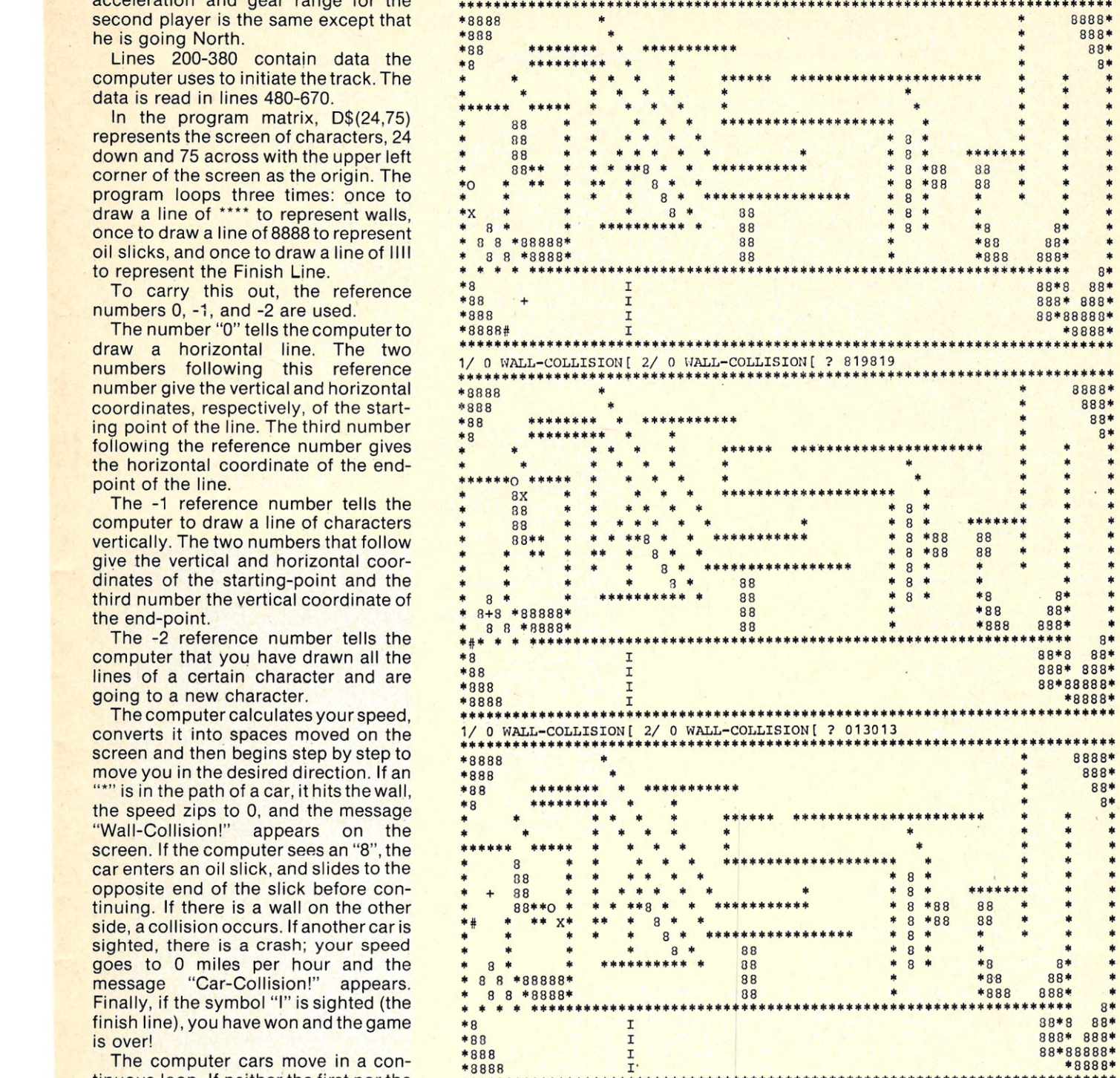
The -2 reference number tells the computer that you have drawn all the lines of a certain character and are going to a new character.

The computer calculates your speed, converts it into spaces moved on the screen and then begins step by step to move you in the desired direction. If an "*" is in the path of a car, it hits the wall, the speed zips to 0, and the message "Wall-Collision!" appears on the screen. If the computer sees an "8", the car enters an oil slick, and slides to the opposite end of the slick before continuing. If there is a wall on the other side, a collision occurs. If another car is sighted, there is a crash; your speed goes to 0 miles per hour and the message "Car-Collision!" appears. Finally, if the symbol "I" is sighted (the finish line), you have won and the game is over!

The computer cars move in a continuous loop. If neither the first nor the second player have won by the time one of the computer cars reaches the finish line, the computer cars will continue on the second lap. You will find that they are average in driving ability.

The data in lines 760-840 describe the points at which the computer cars are to arrive. If the player's car and a computer car land on the same spot, the player's car will disappear for one turn. The computer cars will also stay in their own lane during the race. These cars are represented by the symbols "#" and "+".

```
1/ 100   2/ 100   ? 027027
```



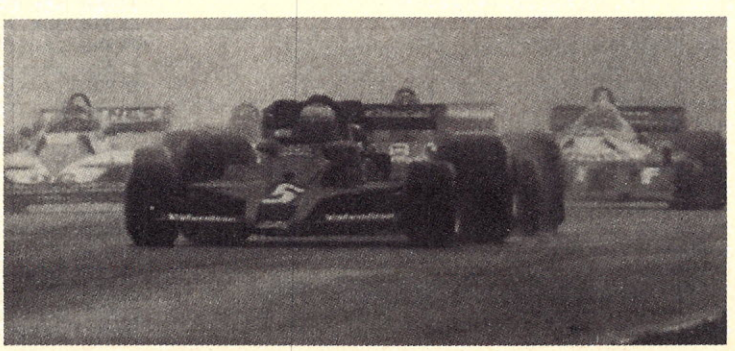
```
1/ 0 WALL-COLLISION[ 2/ 0 WALL-COLLISION[ ? 819819
```



```
1/ 0 WALL-COLLISION[ 2/ 0 WALL-COLLISION[ ? 013013
```

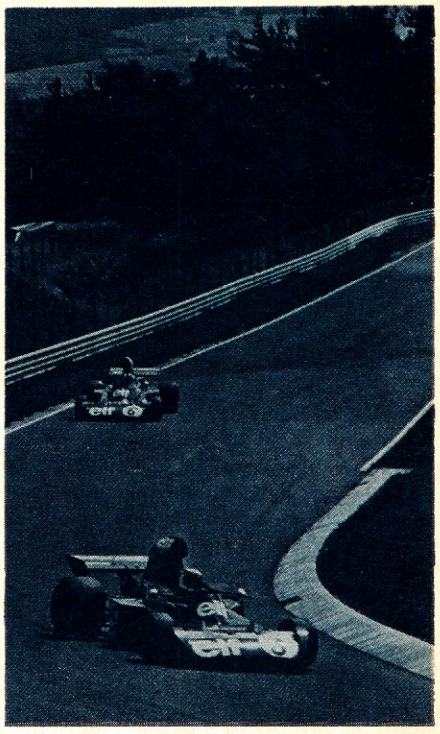


```
1/ 0 WALL-COLLISION[ 2/ 0 CAR-COLLISION[? 812812
```



Program Listing

```
00100 REM              PROGRAM ENTITLED SUPRC "SUPER RACE"
00110 REM            DESIGNED AND WRITTEN BY SCOTT BENNETT IN
00120 REM          CYBER BASIC.  RACE GAME USES TWO CARS.
00130 REM         8'S REPRESENT OIL SLICKS
00140 REM         *'S REPRESENT WALLS
00150 REM         X AND O REPRESENT CARS
00160 REM
00170 REM
00180 DIM D$(24,75),D(2,2),D1(2,2),G(6),C5(6,2),M(38,2)
00190 DIM M1(38,2)
00200 DATA 0,1,1,75,0,4,9,16,0,4,22,32,0,5,9,16,0,6,31,36,0,6,39,60
00210 DATA 0,8,1,6,0,8,9,13,0,9,31,50,0,11,59,65,0,12,30,40,0,14,29,45
00220 DATA 0,16,17,26,0,19,9,70,0,24,1,75,-1,1,1,24,-1,13,6,16,-1,8,13,19
00230 DATA -1,4,16,14,-1,13,20,16,-1,4,25,7,-1,6,31,9,-1,9,50,19
00240 DATA -1,9,54,16,-1,14,60,19,-1,1,65,14,-1,6,70,19,-1,1,75,24
00250 DATA 19,3,2,17,3,18,4,19,5,17,5,20,6,7,6,18,6,21,7,8,7,19,7,22,7,52
00260 DATA8,20,8,23,8,26,8,53,9,21,9,24,9,27,10,20,10,22,10,25,10,28,11,19
00270 DATA11,21,11,23,11,26,11,29,11,40,12,9,12,10,12,18,12,20,12,21
00280 DATA12,24,12,27,13,9,13,10,13,17,13,25,13,28,14,26,15,27,16,28
00290 DATA17,7,18,8,19,3,19,5,19,7,20,69,21,70,22,69,23,70,-2
00300 DATA0,2,2,5,0,2,71,74,0,3,2,4,0,3,72,74,0,4,2,3,0,4,73,74,5,2
00310 DATA5,74,12,55,12,56,12,60,12,61,12,22,13,23,13,55,13,56,13,60
00320 DATA13,61,14,24,15,25,16,4,16,61,16,69,17,3,17,5,0,17,8,12,0
00330 DATA17,61,62,0,17,68,69,18,4,18,6,0,18,9,12,0,18,61,63,0,18,67
00340 DATA69,20,2,20,67,20,68,20,70,20,73,20,74,0,21,2,3,0,21,67,69
00350 DATA0,21,72,74,19,74,0,22,2,4,0,22,67,68,0,22,70,74,0,23,2,5
00360 DATA0,23,71,74,-1,9,7,12,-1,9,8,12,-1,15,33,18,-1,15,34,18
00370 DATA-1,10,52,16,-2
00380 DATA -1,20,20,23,-2
00390 DATA 22,19,20,19
00400 FOR I=1 TO 24
00410 FOR J=1 TO 75
00420 D$(I,J)=" "
00430 NEXT J
00440 NEXT I
00450 Q$(1)="*"
00460 Q$(2)="8"
00470 Q$(3)="I"
00480 FOR Q4=1 TO 3
00490 FOR I=1 TO 100000
00500 READ A
00510 IF A=-2 GOTO 00670
00520 IF A=-1 GOTO 00570
00530 IF A=0 GOTO 00620
00540 READ A1
00550 D$(A,A1)=Q$(Q4)
00560 GOTO 00660
00570 READ A,A1,A2
00580 FOR J=A TO A2
00590 D$(J,A1)=Q$(Q4)
00600 NEXT J
00610 GOTO 00660
00620 READ A,A1,A2
00630 FOR J1=A1 TO A2
00640 D$(A,J1)=Q$(Q4)
00650 NEXT J1
00660 NEXT I
00670 NEXT Q4
00680 MAT READ D
00690 D$(D(1,1),D(1,2))="X"
00700 D$(D(2,1),D(2,2))="O"
00710 MAT D1=D
00720 DATA 100000,10000,1000,100,10,1
00730 MAT READ G
00740 DATA 0,9,1,4,1,9,0,9,1,4,1,9
00750 MAT READ C5
00760 DATA23,13,23,6,19,2,13,2,9,2,9,12,16,12,16,7,8,7,4,4,2,6
00770 DATA2,16,10,24,15,29,15,37,15,46,13,46,13,37,13,29,2,18
00780 DATA2,28,2,44,2,64,10,64,10,58,12,58,12,64,15,64,15,66
00790 DATA2,66,2,70,6,74,18,74,23,69,23,60,23,44,23,26,23,19
00800 DATA21,13,21,8,17,4,11,4,11,11,14,11,14,8,8,8,5,6,5,8,7,10,7,15
00810 DATA17,15,17,29,17,37,17,48,11,48,11,38,11,30,7,26,5,26,5,33
00820 DATA3,35,7,39,7,51,9,53
00830 DATA17,53,17,55,12,60,16,64,12,68,4,68,4,72,17,72
00840 DATA22,68,22,60,22,44,22,19
00850 MAT READ M
00860 MAT READ M1
00870 GOSUB 00900
00880 GOSUB 00980
00890 GOTO 00870
00900 FOR I=1 TO 24
00910 FOR J=1 TO 75
00920 LET A$=D$(I,J)
00930 PRINT A$;
00940 NEXT J
00950 PRINT ""
00960 NEXT I
00970 RETURN
00980 PRINT "1/";S(1);W$(1);" 2/";S(2);W$(2);
00990 W$(1)=" "
01000 W$(2)=" "
01010 INPUT C
01020 FOR I=1 TO 6
01030 FOR J=1 TO 10000000STEP G(I)
01040 D=C-J
01050 IF INT(J/10)=G(I) GOTO 01070
01060 GOTO 01090
01070 G1(I)=9
01080 GOTO 01120
01090 IF D<0 GOTO 01110
01100 NEXT J
01110 G1(I)=INT(J/G(I))-1
01120 C=C-(G1(I)*G(I))
01130 NEXT I
01140 FOR I=1 TO 6
01150 V=C5(I,1)
01160 V1=C5(I,2)
01170 IF G1(I)<VOR G1(I)>V1GOTO 00980
01180 NEXT I
01190 GOSUB 01510
01200 FOR I=1 TO 2
01210 S5(I)=INT(S(I)/10)
01220 NEXT I
01230 FOR I=1 TO 2
01240 ONG1(I*I+2)GOTO 01250, 01460, 01310, 01340, 01480,
      01400, 01370, 01280, 01
01250 D2(I)=1
01260 D3(I)=-1
01270 GOTO 01480
01280 D2(I)=-1*S5(I)
01290 D3(I)=0
01300 GOTO 01480
01310 D2(I)=1
01320 D3(I)=1
01330 GOTO 01480
01340 D2(I)=0
01350 D3(I)=-1*S5(I)
01360 GOTO 01480
01370 D2(I)=-1
01380 D3(I)=-1
01390 GOTO 01480
01400 D2(I)=0
01410 D3(I)=S5(I)
01420 GOTO 01480
01430 D2(I)=-1
01440 D3(I)=1
01450 GOTO 01480
01460 D2(I)=S5(I)
01470 D3(I)=0
01480 NEXT I
01490 GOTO C1960
```



```
01500 REM SPEED/SHIFT
01510 FOR I=1 TO 2
01520 IF S(I)>=0 AND S(I)<=25 AND G1(3*I-1)=1 GOTO 01580
01530 IF S(I)>25 AND S(I)<=100 AND G1(3*I-1)=2 GOTO 01580
01540 IF S(I)>100 AND S(I)<=200 AND G1(3*I-1)=3 GOTO 01580
01550 IF S(I)>200 AND G1(3*I-1)=4 GOTO 01580
01560 S1(I)=1
01570 GOTO 01590
01580 S1(I)=10
01590 IF G1(I*I)>5 GOTO 01760
01600 IF G1(I*I)=0 GOTO 01760
01610 ON G1(I*I) GOTO 01620, 01640, 01660, 01680, 01700
01620 G1(I*I)=60
01630 GOTO 01710
01640 G1(I*I)=75
01650 GOTO 01710
01660 G1(I*I)=50
```

```
01670 GOTO 01710
01680 G1(I*I)=25
01690 GOTO 01710
01700 G1(I*I)=10
01710 S(I)=S(I)-G1(I*I)
01720 IF S(I)<0 GOTO 01740
01730 GOTO 01750
01740 S(I)=0
01750 GOTO 01940
01760 IF G1(I*I)-5=1 GOTO 01810
01770 IF G1(I*I)-5=2 GOTO 01830
01780 IF G1(I*I)-5=3 GOTO 01850
01790 IF G1(I*I)-5=4 GOTO 01870
01800 IF G1(I*I)=0 GOTO 01890
01810 G1(I*I)=10
01820 GOTO C1900
01830 G1(I*I)=25
01840 GOTO C1900
01850 G1(I*I)=100
01860 GOTO C1900
01870 G1(I*I)=200
01880 GOTO C1900
01890 G1(I*I)=1000
01900 S(I)=S(I)+(10*S1(I))
01910 IF S(I)>=G1(I*I) GOTO 01930
01920 GOTO C1940
01930 S(I)=G1(I*I)
01940 NEXT I
01950 RETURN
01960 Z$(1)="O"
01970 Z$(2)="X"
01980 Z1$(1)="X"
01990 Z1$(2)="O"
02000 FOR I=1 TO 2
02010 ON G1(I*I+2) GOTO 02900, 02020, 02900, 02020,
02020 IF D2(I)=0 GOTO 02460
02030 A8=1
02040 IF D2(I)<0 GOTO 02060
02050 GOTO 02070
02060 A8=-1
02070 FOR J=D(I,1) TO (D(I,1)+D2(I)) STEP A8
02080 IF D$(J,D(I,2))="I" GOTO 02130
02090 IF D$(J,D(I,2))="*" GOTO 02190
02100 IF D$(J,D(I,2))="8" GOTO 02250
02110 IF D$(J,D(I,2))=Z$(I) GOTO 02340
02120 GOTO C2320
02130 D$(D1(I,1),D1(I,2))=" "
02140 D(I,1)=J
02150 D$(D(I,1),D(I,2))=Z1$(I)
02160 GOSUB 00900
02170 INPUT A$
02180 STOP
02190 W$(I)="WALL-COLLISION|"
02200 S(I)=0
02210 D$(D1(I,1),D1(I,2))=" "
02220 D(I,1)=J-A8
02230 D$(D(I,1),D(I,2))=Z1$(I)
02240 GOTO 02390
02250 W$(I)="SSKKIDD||||"
02260 FOR F6=A8TO 100*A8 STEP A8
02270 IF D$(J+F6,D(I,2))<>"8" GOTO 02300
02280 NEXT F6
02290 STOP
02300 J=J+F6
02310 GOTO 02080
02320 NEXT J
02330 GOTO 02410
02340 W$(I)="CAR-COLLISION|"
02350 S(I)=0
02360 D$(D1(I,1),D1(I,2))=" "
02370 D(I,1)=J-A8
02380 D$(D(I,1),D(I,2))=Z1$(I)
02390 MAT D1=D
02400 GOTO 02880
02410 D$(D1(I,1),D1(I,2))=" "
02420 D(I,1)=J
02430 D$(D(I,1),D(I,2))=Z1$(I)
02440 MAT D1=D
02450 GOTO 02880
02460 A9=1
02470 IF D3(I)<0 GOTO 02490
02480 GOTO 02500
02490 A9=-1
02500 FOR J1=D(I,2) TO (D(I,2)+D3(I)) STEP A9
02510 IF D$(D(I,1),J1)="I" GOTO 02560
02520 IF D$(D(I,1),J1)="*" GOTO 02620
02530 IF D$(D(I,1),J1)="8" GOTO 02680
02540 IF D$(D(I,1),J1)=Z$(I) GOTO 02770
02550 GOTO 02750
02560 D$(D1(I,1),D1(I,2))=" "
02570 D(I,2)=J1
02580 D$(D(I,1),D(I,2))=Z1$(I)
02590 GOSUB 00900
02600 INPUT A$
02610 STOP
02620 W$(I)="WALL-COLLISION|"
02630 S(I)=0
02640 D$(D1(I,1),D1(I,2))=" "
02650 D(I,2)=J1-A9
02660 D$(D(I,1),D(I,2))=Z1$(I)
02670 GOTO 02820
02680 W$(I)="SSKKIIDD||||"
02690 FOR F6=A9 TO 100*A9 STEP A9
02700 IF D$(D(I,1),J1+F6)<>"8" GOTO 02730
02710 NEXT F6
02720 STOP
02730 J1=J1+F6
02740 GOTO 02510
02750 NEXT J1
02760 GOTO 02840
02770 W$(I)="CAR-COLLISION|"
02780 S(I)=0
02790 D$(D1(I,1),D1(I,2))=" "
02800 D(I,2)=J1-A9
02810 D$(D(I,1),D(I,2))=Z1$(I)
02820 MAT D1=D
02830 GOTO 02880
02840 D$(D1(I,1),D1(I,2))=" "
02850 D(I,2)=J1
02860 D$(D(I,1),D(I,2))=Z1$(I)
02870 MAT D1=D
02880 GOTO 03340
02890 RETURN
02900 FOR J9=1 TO S5(I)
02910 F=F+D2(I)
02920 F1=F1+D3(I)
02930 IF D$(D(I,1)+F,D(I,2)+F1)="I" GOTO 02980
02940 IF D$(D(I,1)+F,D(I,2)+F1)="*" GOTO 03050
02950 IF D$(D(I,1)+F,D(I,2)+F1)="8" GOTO 03120
02960 IF D$(D(I,1)+F,D(I,2)+F1)=Z$(I) GOTO 03190
02970 GOTO 03170
02980 D$(D1(I,1),D1(I,2))=" "
02990 D(I,2)=D(I,2)+F1
03000 D(I,1)=D(I,1)+F
03010 D$(D(I,1),D(I,2))=Z1$(I)
03020 GOSUB 00900
03030 INPUT A$
03040 STOP
03050 W$(I)="WALL-COLLISION|"
03060 S(I)=0
03070 D$(D1(I,1),D1(I,2))=" "
03080 D(I,2)=D(I,2)+(F1-D3(I))
03090 D(I,1)=D(I,1)+(F-D2(I))
03100 D$(D(I,1),D(I,2))=Z1$(I)
03110 GOTO 03250
03120 W$(I)="SSKKIIDD||"
03130 F=F+D2(I)
03140 F1=F1+D3(I)
03150 IF D$(D(I,1)+F,D(I,2)+F1)<>"8" GOTO 02930
03160 GOTO 03130
03170 NEXT J9
03180 GOTO 03270
03190 W$(I)="CAR-COLLISION|"
03200 S(I)=0
03210 D$(D1(I,1),D1(I,2))=" "
03220 D(I,1)=D(I,1)+(F-D2(I))
03230 D(I,2)=D(I,2)+(F1-D3(I))
03240 D$(D(I,1),D(I,2))=Z1$(I)
03250 MAT D1=D
03260 GOTO 03320
03270 D$(D1(I,1),D1(I,2))=" "
03280 D(I,1)=D(I,1)+F
03290 D(I,2)=D(I,2)+F1
03300 D$(D(I,1),D(I,2))=Z1$(I)
03310 MAT D1=D
03320 F1=0
03330 F=0
03340 NEXT I
03350 GOSUB 03370
03360 RETURN
03370 P5=P5+1
03380 IF P5=1 GOTO 03440
03390 D$(M(P5-1,1),M(P5-1,2))=" "
03400 D$(M1(P5-1,1),M1(P5-1,2))=" "
03410 IF P5=38 GOTO 03430
03420 GOTO 03440
03430 P5=1
03440 D$(M(P5,1),M(P5,2))="#"
03450 D$(M1(P5,1),M1(P5,2))="+"
03460 RETURN
03470 END
```

02020, 02020, C2900, 02020,

IF YOU'RE NOT SUBSCRIBING TO CREATIVE COMPUTING, YOU'RE NOT GETTING THE MOST OUT OF YOUR COMPUTER.

No computer magazine gives you more applications than we do! Games. Puzzles. Sports simulations. CAI. Computer art. Artificial intelligence. Needlepoint. Music and speech synthesis. Investment analysis. You name it. We've got it. And that's just the beginning!

Whatever your access to computer power—home computer kit, mini, time-sharing terminal—*Creative Computing* is on your wavelength. Whatever your computer application—recreation, education, business, household management, even building control—*Creative Computing* speaks your language.

Read through pages of thoroughly documented programs with complete listings and sample runs. All made easy for you to use. Learn about everything from new software to microprocessors to new uses for home computers. And all in simple, understandable terms. And there's still more. *Creative Computing* discusses creative programming techniques like sort algorithms, shuffling and string manipulation to make your own programming easier and more efficient.

We can even save you time and money. Our extensive resource section is filled with all kinds of facts plus evaluations of hundreds of items. Including microcomputers, terminals, peripherals, software packages, periodicals, booklets and dealers. We also give you no-nonsense equipment profiles to help you decide which computer is best for you—before you spend money on one that isn't.

We've got fiction too. From the best authors in the field, like Asimov, Pohl and Clarke. Plus timely reviews of computer books, vendor manuals and government pamphlets. And so much more!

Isn't it about time you subscribed to *Creative Computing*? It's the smart way to get the most out of your computer.

Complete this coupon and mail it today. Or for fast response, call our toll-free hot line. **(800) 631-8112**. (In New Jersey call (201) 540-0445).

I want to get the most out of my computer.
Please enter my subscription to:

creative computing

Term	USA	Foreign Surface	Foreign Air
☐ 1-year	☐$ 8	☐$ 12	☐$ 20
☐ 2-year	☐$ 15	☐$ 23	☐$ 39
☐ 3-year	☐$ 21	☐$ 33	☐$ 57
☐ Lifetime	☐$300	☐$400	☐$600
☐ Vol. 1 Bound	☐$ 10	☐$ 12	☐$ 15
☐ Vol. 2 Bound	☐$ 10	☐$ 12	☐$ 15

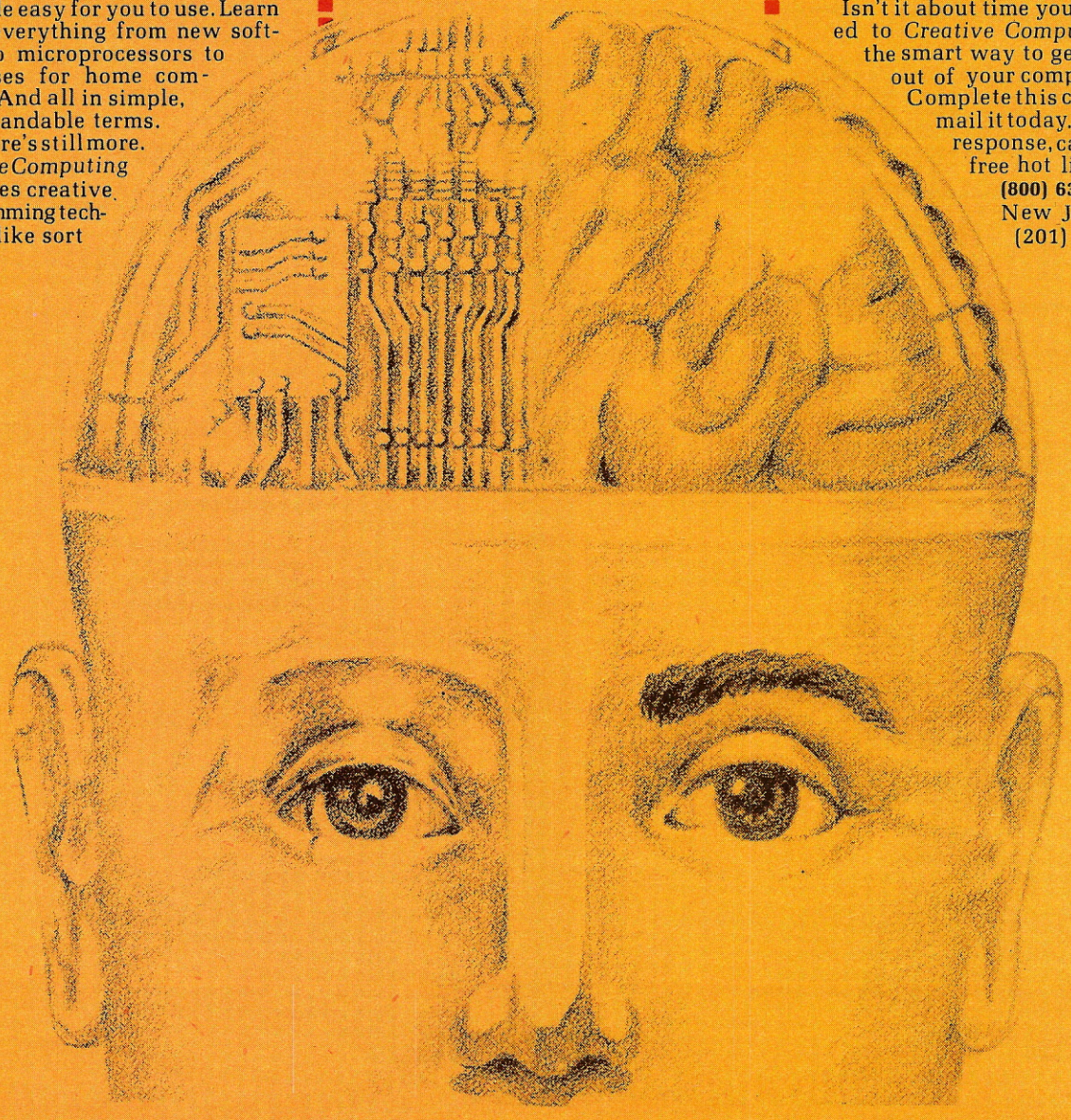
☐ Payment Enclosed
☐ Visa/Bank Americard ☐ Master Charge
Card No. _____
☐ Please bill me ($1.00 billing fee will be added; foreign orders must be prepaid)

Name _____

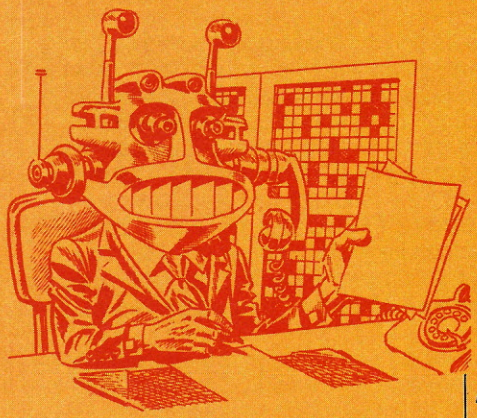
Address _____

City _____ State _____ Zip _____

Send to: Creative Computing, Atten: Emma
P.O. Box 789-M, Morristown, N.J. 07960



creative computing brings you its best!



Creative Computing Magazine

The #1 computer applications magazine helps you get the most out of your computer, be it a micro, mini or timesharing system. Plenty of practical applications in education, small business management, household management, recreation and games, and building control. Programming techniques, for those writing their own, include sort techniques, recursion, and file structures. Complete listings and sample runs of games, investment analysis, music synthesis, artificial intelligence programs and much more. Articles, in-depth book and resource reviews, hardware and software evaluations and even some fiction and foolishness. "But the best thing about *Creative Computing* is its feisty, friendly attitude" — *The Village Voice.* Subscriptions, USA 1 year $8 [1A], 3 years $21 [1B]; foreign 1 year $12 [1G], 3 years $33 [1H]. Sample copy $2 [1C]. Volume 1 and 2 issues have been edited into two big 324-page books. *American Vocational Journal* said of Volume 1, "This book is the 'Whole Earth Catalog' of computers." $8.95 [6A]. Volume 2 continues in the same tradition, "Non-technical in approach, its pages are filled with information, articles, games, and activities. Fun layout." — *American Libraries.* $8.95 [6B]. Also available is a special package of all the single issues in Volume 3 for only $8 [1N].

Some Common BASIC Programs

Adam Osborne. An ideal workbook to accompany a BASIC programming course or for one's own use. Contains 76 general purpose, practical BASIC programs with complete listings. 192 pp. $7.50 [7M].

Game Playing With Computers

Donald Spencer. A good mind-exercising resource. Contains over 70 games, puzzles and recreations for the computer. The games, in both BASIC and FORTRAN, have detailed descriptions, flowcharts and output. Also includes a history of game-playing machines and lots of "how-to" write your own games. 320 pp. hardbound $16.95 [8S].

BASIC Programming

Kemeny and *Kurtz.* An excellent gradual introduction to computer programming in BASIC with lots of samples ranging from simple to complex. Rated "The best text on BASIC on almost all counts." 150 pp. hardbound $8.50 [7E].

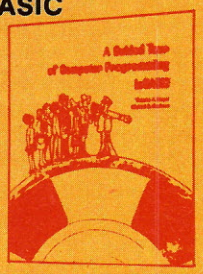
Byte Magazine

If you're considering your own personal computer, then you need *Byte.* With an emphasis on hardware, *Byte* provides a wealth of information on how to get started at an affordable price. Theory, opinion, how-to articles. Subscriptions, USA 1 year $12 [2A], 3 years $32 [2B]; foreign 1 year $25 [2F]. First issues of this now classic publication have been edited into *The Best of Byte, Volume 1.* 386 pp. $11.95 [6F].

Microcomputer Design

Donald P. Martin. This book is well-suited for the engineer who's designing microprocessors into his company's products. Not just block diagrams or vague theory, but dozens of practical circuits with schematics for CPUs based on 8008 chips. Includes interfacing to A/D, D/A, LED digits, UARTs, teletypewriters. Over 400 pp $14.95 [9P].

A Guided Tour of Computer Programming In BASIC



Dyer and *Kaufman.* This book tops all introductory texts on BASIC. Filled with detail and examples, it includes sample programs for many simulations, several games, reservations systems and payroll. Aimed at the novice, but of value to everyone. 156 pp. $4.80 [8L].

For faster service call toll free

800-631-8112

In NJ call **201-540-0445**

Artist And Computer

"Get yourselves a copy of this book if you enjoy feeding your mind a diet of tantalizing high-impact information." — *San Francisco Review of Books.* In this book, 35 artists present a multitude of computer uses and the very latest techniques in computer-generated art. 132 pp. $4.95 softbound, [6D], $10 hardbound [6E].

BASIC Computer Games: Microcomputer Edition

David H. Ahl. An anthology of 101 games and simulations. Everything from Acey-Deucey to Yahtzee, all in BASIC. The only book of its kind with complete listings, sample runs and descriptive write-up of each game. 248 pp. $7.50 [6C].

Creative Computing Catalog

FREE 16-page reference to computer-related products not readily available in commercial stores. Lists books, magazines, T-shirts, game, binary dice and more [5A

The Colossal Computer Cartoon Book

Sit back and relax. Take a break with the best collection of computer cartoons ever. Popular gift item, too. 128 pp. $4.95 [6G].

Problem Solving With The Computer

Ted Sage. Used in conjunction with the traditional high school math curriculum, this book stresses problem analysis in algebra and geometry. This is the most widely adopted text in computer mathematics. 244 pp. $6.95 [8J].

Computer Rage

Fun and educational new board game based on a large-scale multi-processing computer system. The object is to move your three programs from input to output. Moves are determined by the roll of three binary dice representing bits. Hazards include priority interrupts, program bugs, decision symbols, power failures and restricted input and output channels. Notes included for adapting game for school instruction. Great gift item. Ages 8-adult, 2-4 players. $8.95 [6H].

Getting Involved With Your Own Computer

Solomon and *Viet.* One of the first books on microcomputers that requires no previous knowledge of electronics or computer programming. Tells you where to find information, explains basic concepts and summarizes existing systems. Good place for the neophyte to begin. 216 pp. $5.95 [9N].

Fun & Games With The Computer

Ted Sage. "This book is designed as a text for a one-semester course in computer programming using the BASIC language. The programs used as illustrations and exercises are games rather than mathematical algorithms, in order to make the book appealing and accessible to more students. The text is well written, with many excellent sample programs. Highly recommended." — *The Mathematics Teacher.* 351 pp. $6.95 [8B].

An Introduction to Microcomputers

Adam Osborne. Volume 0, "The Beginner's Book" covers microcomputer system components, how they work together; number systems, the basics of programming, and putting it all together in a system of your own. 264 pp. $7.50 [9T]. Volume 1, Basic Concepts, also assumes no prior knowledge of computers. It covers basic principles, binary arithmetic, the microprocessor CPU, I/O logic, memory organization and programming. 264 pp. $7.50 [9K]. Volume 2 (2nd Edition), Some Real Products, covers 20 actual microprocessors in considerable detail including timing diagrams, instruction sets, and interrupts. 760 pp. $15 [9L].

creative computing P.O. Box 789M, Morristown, NJ 07960 USA

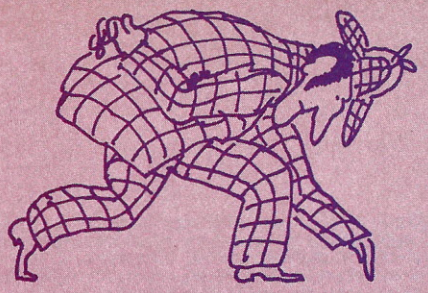
Please send me the following:

Quan.	Cat.	Description	Price

Books shipping charge
$1 USA,$2 Foreign _____
NJ residents add 5% Sales tax _____
8 UVW TOTAL _____

Name _____
Address _____
City _____ State _____ Zip _____
□ Cash, check, M.O. enclosed
□ Bank Americard/VISA
□ Master Charge
Card No. _____
Expiration date _____

CIRCLE 124 ON READER SERVICE CARD



Index to Advertisers

Reader Service No.	Advertiser	Page No.
149	Addison-Wesley Publishing Co.	87
150	Ancrona Corporation	27
151	Apollo	49
132	Byte Shop of San Jose	129
131	Carterfone Communications	91
119	Computalker Consultants	54
152	Computer Components	131
133	Computer Corner	49
153	Computer Data Systems	11
134	Computer Depot Inc	52
154	Computers Etc.	16
135	Computer Enterprises	85
155	Computerfest '78	102
136	Computer Hardware of Virginia	129
137	Computer Mart of California	53
127	Computer Mart of New Jersey	114
126	Computer Mart of New York	129
138	Computers Plus, Inc.	129
156	Continental Specialties Corp	7
139	Corson Computer Corp.	52
124	Creative Computing	134, 142, 143
102	Cromemco	1
104	Cybercom/Solid State Music	35
131	Dal-Data Inc.	91
123	Electrolabs	127
114	E & L Instruments	CIII
122	Four Corners Press	50
105	Heath Co.	5
140	Home Computer Center of Virginia	129
157	McGraw Hill Book Company	43
158	Micro Expo Paris	110
141	Micro Logistics	50
	Mini/Micro '78	105
106	Miniterm Associates	9
160	NCC '78 Festival	115
142	Netronics R & D	13
117	OK Machine and Tool Company	22, 23
118	Osborne & Associates	51
161	Oxford Software Corp.	18
115	Parasitic Engineering	17
162	Percomp '78	38, 39
143	Personal Computer Corp.	129
	PolyMorphic Systems	CIV
107	Processor Technology	15
163	Smoke Signal Broadcasting	19
164	SOROC Technology	41
101	Southwest Technical Products Co.	CII
144	Summagraphics	94
165	Sunshine Computer Company	21
128	Sybex	82, 83
120	Tarbell Electronics	53
116	Tektronix	2
145	Telecom	50
146	The Electronics Place	129
	Trenton Fair	121
147	Wameco, Inc.	54, 55
125	Windjammer	95
148	Xybex	55

MORE Information MORE

At the right price to you, the reader (namely a 13¢ stamp), *Creative Computing* is providing a new service to help you get all the computer product and service information for which you used to have to spend laborious time writing to each company, finding misplaced stamps, and looking up addresses. No more! You need only circle the numbers on the reader service card to the right that correspond to the companies from which you'd like more information, find *one* stamp, drop it in the mail, and wait for your postman. *Creative Computing* does the rest.

We hope this new service helps us too. Increasing numbers of subscribers are writing *Creative Computing* asking us for information on new products and services they read about each issue. It's impossible for us to stock manufacturer literature, especially in the microcomputer field where state-of-the-art development is so rapid. Instead, we're making it easy for you to get material directly from the sources involved.

Creative Computing is pleased to bring you this speedy, economical way to get information and hopes you feel it is a worthwhile addition to everybody's favorite magazine.

Coming in May

- **Special Section on Animation, Art and Graphics.** Here's what's happening at the state-of-the-art forefront in computer art, covering the latest in the complexities of animation, the advances in computer-generated art, and where it's at with graphics, including the use of a light pen to interact with color graphics.
- **Business Computing: Payroll.** The second part of this new series will take a look at several of the payroll programs available for small-business use of personal computers. As in the issue you have in your hands, there will be an independent overview, followed by descriptions of half a dozen payroll programs, from a variety of sources, including hardware manufacturers, software houses, and computer stores.
- **What's a Programmer?** Although located in the fiction department of the magazine, this story was obviously written by a man with much experience as "the guy who writes the instructions that tell the big black box to overcharge you by seventeen cents, so you'll spend ten dollars on phone calls and gas getting the seventeen cents off your bill." He tells exactly why most of such problems occur, sometimes through customer error, sometimes because a computer ... but why not get the May-June issue and find out for yourself?
- **Games, Games, Games.** A couple of new ones you'll rush to feed into your computer as soon as you see them. Complete listings, runs and descriptions, of course.
- **What the Computer Taught Me About My Students.** Subtitled, Is Binary Search "Natural"? A college teacher tells what happened when a freshman class was assigned to write a program to play the game, "I'm thinking of a number."
- **If It's Right, You Know It.** This is a brief look at what several young people think of their careers in data processing. Most are quite enthusiastic about working with computers, while some others....

Free Bugbacks.

$6.50 worth of Bugback® labels free with $35.00 book purchase.

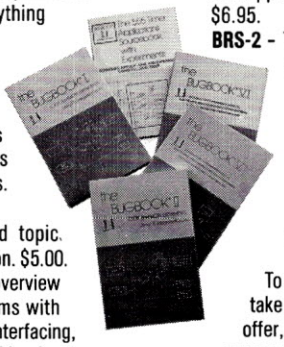
Our Bugback adhesive labels stick to their chips, immediately identifying logic functions and pin numbers. But even without the $6.50 Bugback freebie, our Bugbooks® are a great bargain. Nine books, packed with everything to teach you digital and analog electronics from scratch.

Bugbooks I & II - 750 pages, with experiments in basic digital electronics to sophisticated circuits. $17.00.

Bugbook IIA - Advanced topic. in digital logic transmission. $5.00.

Bugbook III - General overview of microcomputer systems with particular emphasis on interfacing, I/O, programming in machine languages. 597 pages. $15.00.

Bugbooks V & VI - Digital logic and microcomputers from ground zero. Emphasis on self-teaching. No previous knowledge of electronics needed. Over 900 pages. $19.00.



BRS-1 - Lab workbook providing theory and applications of the 555 timer chip. $6.95.

BRS-2 - Text/workbook covering the best design of low-pass, high-pass, bandpass and notch filters. $8.50

BRS-4 - OP-AMP and linear course. 300 pages. $9.00.

Select your $35 (or more) book value and get $6.50 worth of Bugbacks now.

To order your books and take advantage of this special offer, contact one of our representatives listed below.

E&L®

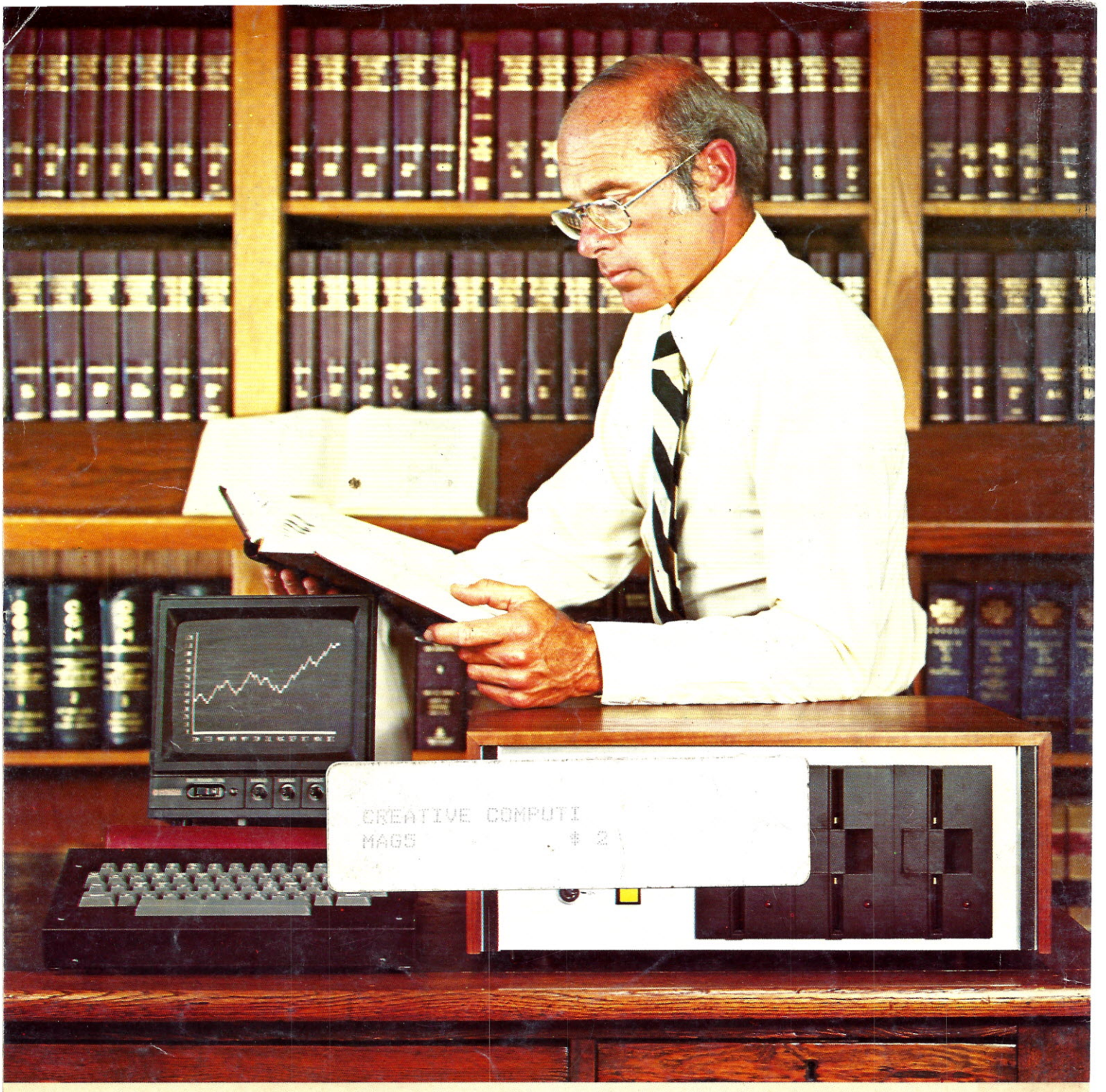
E&L INSTRUMENTS, INC.
61 First Street, Derby, Conn.. 06418
(203) 735-8774 Telex No. 96 3536

Authorized Stocking Representatives

Los Angeles, Calif. (213) 328-9770	Kansas City, Kansas (913) 649-8952	Dallas, Texas (214) 328-5484
San Francisco, Calif. (415) 961-5500	Minneapolis, Minn. (612) 927-5601	Fairfax, Virginia (703) 273-1803
Woodbridge, Conn. (203) 397-1461	New York, N.Y. (201) 467-8585	Seattle, Washington (206) 938-4166
Denver, Colorado (303) 534-1356	Syracuse, N.Y. (315) 699-2651	
Orlando, Florida (305) 351-1841	Dayton, Ohio (513) 222-0011	
Chicago, Illinois (312) 956-8090	Philadelphia, Pa. (215) 723-8733	

TM

CIRCLE 114 ON READER SERVICE CARD



The Computer for the Professional

Whether you are a manager, scientist, educator, lawyer, accountant or medical professional, the System 8813 will make you more productive in your profession. It can keep track of your receivables, project future sales, evaluate investment opportunities, or collect data in the laboratory.

Use the System 8813 to develop reports, analyze and store lists and schedules, or to teach others about computers. It is easily used by novices and experts alike.

Reliable hardware and sophisticated software make this system a useful tool. Several software packages are included with the machine: an advanced disk operating system supporting a powerful BASIC language interpreter, easy to use text editor, assembler and other system utilities. Prices for complete systems start at $3250.

See it at your local computer store or contact us at 460 Ward Dr., Santa Barbara, CA 93111, (805) 967-0468.

PolyMorphic Systems